



ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ  
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ  
УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ

МАСТЕР РАД

---

**Моделовање тема у систему за предлагање  
одговора на питања постављена природним  
језиком**

---

*Студент:*  
Јелица Васиљевић

*Професор:*  
др Милош Ивановић

Август 2015.

# Садржај

<b>1</b>	<b>Увод</b>	<b>3</b>
1.1	Опис проблема . . . . .	3
1.2	Опис решења . . . . .	3
1.3	Терминологија . . . . .	3
1.4	Коришћене скраћенице . . . . .	4
<b>2</b>	<b>Алгоритам моделовања тема</b>	<b>5</b>
2.1	<i>Latent Dirichlet allocation</i> . . . . .	6
2.1.1	Генеративни процес . . . . .	7
2.1.2	Како се откривају теме - упрошћен пример . . . . .	8
<b>3</b>	<b>Математичка позадина</b>	<b>13</b>
3.1	Теорија вероватноће . . . . .	13
3.1.1	Основни појмови . . . . .	13
3.1.2	Условна вероватноћа . . . . .	14
3.1.3	Случајне променљиве . . . . .	16
3.1.3.1	Дискретне случајне променљиве . . . . .	17
3.1.3.2	Непрекидне случајне променљиве . . . . .	19
3.1.3.3	Вишедимензионалне случајне променљиве . . . . .	20
3.2	Важније расподеле . . . . .	24
3.2.1	Биномна и полиномна(енг. multinomial ) расподела . . . . .	24
3.2.2	Дирихлеова расподела . . . . .	25
3.3	Гибсово узорковање . . . . .	27
3.3.1	Марковљеви ланци . . . . .	27
3.4	Како ради ТМ алгоритам . . . . .	30
3.4.1	Имплементација - псеудокод . . . . .	34
<b>4</b>	<b>Припрема подарака</b>	<b>35</b>
4.1	Уклањање HTML ознака и неалфанумеричких карактера . . . . .	35
4.2	Конвертовање свих слова текста у „мала слова“ енгл. lowercase . . . . .	36
4.3	Издавање атомских елемената докумената - токена, енгл. tokenization . . . . .	36
4.4	Избацивање често коришћених речи енгл. stop words . . . . .	36
4.5	Додавање синонима . . . . .	37
4.6	Склањање наставака речи - енгл. stemming . . . . .	37
4.7	Свођење на коренску реч - енгл. lemmatization . . . . .	38
<b>5</b>	<b>Решење проблема применом алгоритма моделовања тема</b>	<b>39</b>
5.1	Опис решења . . . . .	40
5.2	Мерење сличности . . . . .	41
5.2.1	Косинусна сличност . . . . .	41
5.2.2	Мерење сличности према лексичкој и тематској сличности . . . . .	42
5.2.3	Мерење сличности према предвиђеној вероватноћи . . . . .	43

<b>6</b>	<b>Развој решења</b>	<b>46</b>
6.1	Општи преглед пакета Mallet . . . . .	46
6.1.1	Подаци у Mallet-у . . . . .	46
6.1.2	Алгоритам моделовања тема у Mallet-у . . . . .	47
6.2	Опис решења . . . . .	48
<b>7</b>	<b>Решење проблема применом методе бројања речи</b>	<b>50</b>
7.1	Опис решења методом бројања речи . . . . .	50
<b>8</b>	<b>Преглед резултата</b>	<b>52</b>
8.1	Утицај броја тема и броја итерација на просечну позицију . . . . .	52
8.2	Утицај корака предпроцесирања на просечну позицију . . . . .	53
8.2.1	Резултати без додатних трансформација . . . . .	53
8.2.1.1	Косинусна сличност . . . . .	54
8.2.1.2	Мерење сличности према лексичкој и тематској сличности . .	54
8.2.2	Утицај стеминга на резултат . . . . .	55
8.2.2.1	Косинусна сличност . . . . .	55
8.2.2.2	Мерење сличности према лексичкој и тематској сличности . .	56
8.2.3	Утицај лемитизације на резултат . . . . .	56
8.2.3.1	Косинусна сличност . . . . .	56
8.2.3.2	Мерење сличности према лексичкој и тематској сличности . .	57
8.2.4	Утицај додавања синонима на резултат . . . . .	58
8.2.4.1	Косинусна сличност . . . . .	58
8.2.4.2	Мерење сличности према лексичкој и тематској сличности . .	59
8.2.5	Укупни резултати са синонимима и стемингом . . . . .	60
8.2.5.1	Косинусна сличност . . . . .	60
8.2.6	Укупни резултати са синонимима и лемитизацијом . . . . .	61
8.2.6.1	Косинусна сличност . . . . .	61
8.2.6.2	Мерење сличности према лексичкој и тематској сличности . .	62
8.3	Упоредни резултати решења алгоритмом моделовања тема и бројањем речи .	63
8.3.1	Косинусна сличност . . . . .	63
8.3.2	Мерење сличности према лексичкој и тематској сличности . . . . .	63
8.3.3	Сличност према предвиђеној вероватноћи . . . . .	64
<b>9</b>	<b>Додатак</b>	<b>69</b>
9.1	Предпроцесирање . . . . .	69
9.2	Оптималан број тема и итерација . . . . .	70
9.3	Мерење сличности . . . . .	70
<b>10</b>	<b>Литература</b>	<b>72</b>

# Глава 1

## Увод

### 1.1 Опис проблема

Проналажење одговора на постављено питање представља свакодневни проблем. Сваким Гугл упитом, покреће се низ алгоритама коју покушавају да одгонетну шта упит заправо представља, које би странице биле релевантне и у ком редоследу. Мера „доброг“ одговора на постављено питање знатно може да варира у зависности од сврхе система. Интуитивно, тематика одговора и питања може да послужи као добар критеријум одабира квалитетних одговора. Задатак овог рада је испитивање да ли заиста тематика може да помогне у проналажењу адекватног одговора и у коликој мери.

### 1.2 Опис решења

Циљ овог рада је израда прототипа програма који би коришћењем алгоритама за моделовање тема из базе потенцијалних одговора проналазио најбољи одговор за задато питање. Дакле, програм не треба да „осмисли“ одговор на задато питање већ само да „препозна“ који од могућих одговора највише одговара постављеном питању.

Примена оваквог решења могла би да буде значајна у различитим областима од комерцијалних до научних. На пример, омогућило би се ефикасно аутоматско одговарање на често постављена питања која могу имати различиту формулацију или ефикасно проналажење адекватних научних радова.

У раду су обрађивани текстови на енглеском језику али због природе модела, развијени програм и резултате могуће је применити и на било који други језик. Поред основног текста питања и одговора, у раду је испитан и утицај додавања синонима на проналажење одговора као и утицај свођења речи на коренске (склањање глаголских и именских наставака, енг. *stemming*).

Као компаративни модел коришћен је приступ проналажења одговора на основу броја заједничких речи (енг. *WordCount*).

### 1.3 Терминологија

Општи преглед значења термина који су коришћени у раду као што су реч, речник, вокабулар, корпус итд.

- Тема : скуп речи које је најбоље карактеришу. На пример тема рачунарство би представљала скуп речи : алгоритам, процесор, кодирање, израчунавање, меморија, рачунар, бит, бајт, лаптоп итд. Важно је приметити да неке речи могу припадати у више тема, као што је нпр. реч израчунавање која може припадати и области математика.

- Речник или корпус - скуп свих различитих речи које се јављају у неком скупу докумената
- 

## 1.4 Коришћене скраћенице

1. енг. - енглески језик
2. срп. - српски језик
3. LDA - Latent Dirichlet Allocation
4. ТМ - Topic Model, срп. моделовање тема

## Глава 2

# Алгоритам моделовања тема

Сваким даном повећава се количина доступних дигиталних информација. Парадокс данашњег времена је да се упркос великој количини података из различитих области све теже долази до података који су од интереса. Дакле, потребно је пронаћи алат којим би се велике количине података организовале а самим тим боље разумеле и лакше претреживале.

Тренутно, најпопуларнији начин претреге је према кључним речима. Кључне речи се предају неком систему за претрагу а као резултат добијамо скуп докумената који су повезани са њима. Иако овакав систем ради јако добро и са великом поузданошћу, постоје и другачији приступи

Најчешће питање које се поставља захтева одговр из неколико, ужих или ширих области. Међутим, кључне речи које се предају као критеријум претраге могу карактерисати и области које нису од интереса. На пример, Гугл претрга за кључне речи "ген, еволуција" ће у највећем броју случајева садржати веб странице посвећене биологији или сродним областима. Међутим, поменуте речи припадају и области рачунарства ( генетски алгоритми ) али ти резултати ће знатно слабије бити заступљени у односу на резултате везане за биологију. Тренутно се оваква врста проблема може решити додавањем још неке кључне речи која припада захтеваној области ( на пример за задате речи : алгоритам, кодирање, програм итд. ) међутим поставља се питање избора адекватних додатних речи, њихове заступљености у захтеваној области као и релевантности у односу на друге кључне речи ( на примеру гугл претраге, често се дешава да механизам за претраживање испусти неку од речи и прикаже само резултат за остатак упита )

Један од начина да се поменути проблем реши је и претрага по тематици или теми докумената. Дакле, уместо претраге по кључним речима, најпре се пронађе област у којој се врши претраживање а затим се претражују документи који припадају тој области. На тај начин елиминишу се документи који би се у резултату појавили на основу предатих кључних речи али који нису повезани са тематиком у којој се тражи одговор.

На пример, нека је потребно пронаћи све чланке листа "Политика" у којима се говори о спортским успесима наше кошаркашке репрезентације. Сви чланци овог листа могу се поделити у неколико категорија : политика, хроника, спорт, време итд. Пошто је од интереса категорија спорта, у обзир претраге долазе једино спортски чланци са тематиком кошарке. На овај начин могуће је пратити како се мењао успех кошаркашке репрезентације са временом, колико се обаћала пажња на такве догађаје у ком временском периоду итд.

Описани вид претраге подразумевао би да се сваки лист "Политике" најпре прочита а затим раздвоји по категоријама и "разуме" шта која категорија представља како би се добили тражени подаци. Због велике количине података, овако нешто је немогуће урадити без помоћи рачунара. Алгоритми моделовања тема представљају први корак ка решавању оваквих и сродних проблема.

Циљ алгоритама за моделовање тема је "отривање" тема присутних у некој колекцији докумената. У суштини, алгоритми за моделовање су статистичке методе које на основу

анализе свих речи у документу откривају које су то теме заступљене у том документу. За рад ове методе није потребно никакво претходно означавање докумената, теме докумената зависе једино од речи које се јављају у тексту.

Моделовање тема омогућава организацију велике количине података на нивоу који је тешко дохватљив људским могућностима.

Надаље ће се паралелно употребљавати две еквивалентне ознаке - моделовање тема или ТМ ( скраћеница од енг. *topic model*)

## 2.1 *Latent Dirichlet allocation*

*Latent Dirichlet allocation*, надаље LDA, је најједноставнији приступ проблему моделовања тема [2] а његова примена била је и предмет овог рада.

Добро је познат роман Бранка Ћопића "Орлови рано лете". Уколико би неко ко није прочитао ови књигу желео да зна "о чему се ради"у њој, највероватније би добио одговор да је у питању књига која се бави доживљајима групе дечака на почетку Другог светског рата. Иако је то најшири оквир романа,у њему су присутне и теме о љубави, дружењу, пријатељству, рату, пустоловинама итд. Дакле, роман ,опште гледано, обухвата више тема, али се са неколико њих интензивно бави.

Уколико сада посматрамо овај рад као један пример документа, у њему се највише "говори"о моделовању тема и њиховој примени, али исто тако, само у мањој мери, и о књижевности ( пример књиге "Орлови рано лете "). Дакле, тешко је пронаћи било какав документ који се бави само једном темом. Чак се и у радовима који се баве неким уским научним областима, могу пронаћи делови који се могу сврстати и у неке друге научне области, било из исте, сродне или потпуно различите научне гране.

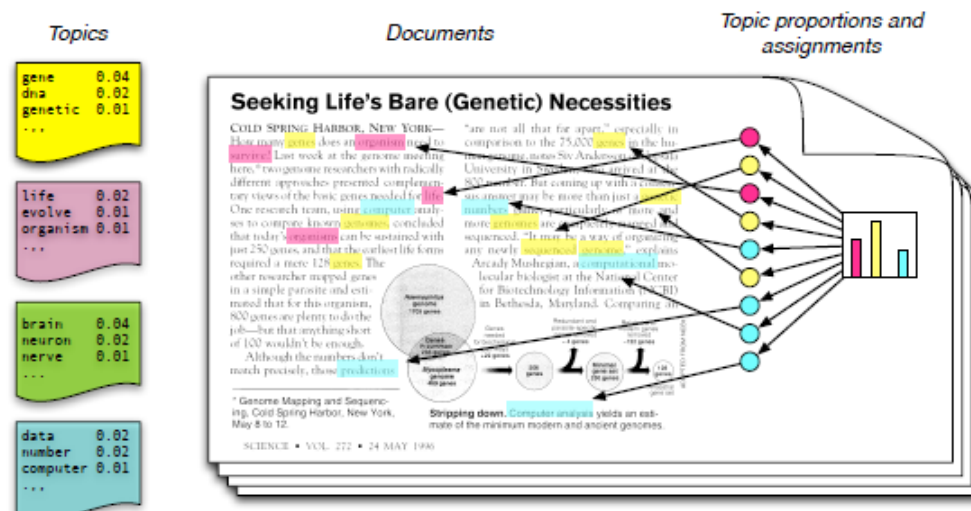
Основна претпоставка LDA модела је управо да се сваки документ може сврстати у више области, тј. да се бави са неколико тема.

Поставља се питање како то, да након читања неког документа, знамо да су у документу присутне различите теме и које су то теме. Уколико неки рад који се бави тематиком машинском учења прочита човек који се не бави рачунарством, највероватније је да би он као теме рада издвојио рачунарство и математику. Међутим, ако би исти рад прочитао неко ко се бави рачунарством, тада би он могао као потенцијалне теме да изведе једну или више области машинског учења, алгоритме, базе података, статистику, теорију бројева итд. Дакле, издвојене теме зависе од „знања“ онога ко чита рад. Уопштено, у оба случаја теме су издвојена на основу речи које се најчешће срећу у областима које појединац познаје. Дакле, може се уопштити да тема представља **скуп карактеристичних речи**.

На следећој слици представљен је чланак "*Seeking Life's Bare(Genetic) Necessities*" који „говори о“ употреби анализе података за одређивање броја гена који организам треба да има да би преживео ( у еволутином смислу). Може се уочити да су три најзаступљеније области у овом тексту - анализа података, еволутивна биологија и генетика. На њој су "ручно"означене неке речи које припадају овим областима. Речи које се могу сврстати у област *анализе података* су означене плавом бојом, речи које припадају *генетици* су означене жутом бојом док су речи које се односе на *еволутивну биологију* означене розом бојом. Уколико би се ова процедура применила на сваку реч текста, јасно би се уочило колико је која тема заступљена у овом тексту. Математички, "*присуство*"теме у тексту се означава односом броја "обојених"речи у једну боју и укупног броја речи у тексту.

Наравно, постоје речи које се могу сврстати у више од једне теме. Такве речи ће бити обојене са две или више боја,али због прегледности слике, такви случајеви су изостављени.

На левој стани слике дате су неке теме ( енг. *Topics*) које са одређеним вероватноћама садрже неке речи. На основу припадности речи темама, извршава се описани процес означавања ("бојења") речи да би се на крају добили удели тема у тексту ( десна страна



Слика 2.1: Пример чланка, преузето са [2]

слике, енг. *Topics* ) Важно је приметити да у тексту постоји доста речи које не одређују ни једну конкретну тему и које се са готово истим, малим вероватноћама могу сврстати у сваку од њих. Такве речи су нпр. везници, личне заменице ,прилошке одредбе итд. Оне се једним именом називају енг. *stop words*. Како присуство таквих речи не утиче на тематику документа, то их при математичкој анализи текста треба занемарити.

LDA је статистички модел који формално описује описани процес означавања речи. Да би се у потпуности разумело како LDA ради, потребно је упознати се са *генеративним процесом* - процесом којим се креирају документи са становишта LDA-а.

### 2.1.1 Генеративни процес

Нека је дат неки скуп речи најчешће коришћених у научним областима математике, физике, хемије, музике, рачунарства и биологије. Нека је, даље, потребно креирати документ који има одређен број речи из датог скупа речи али тако да он највише "говори о"математици и музици, али поред ових тема, у мањој мери, "говори о"физици и програмирању, док се остале теме занемарљиво мало помињу. Како је дат фиксан скуп речи - речник (вокабулар), могуће је свакој речи придружити *вероватноћу* припадања свакој од тема. Тако ће на пример реч *интеграл* имати велике вероватноће припадања темама математика и физика, мање вероватноће у темама хемија и рачунарство док ће се са јако малим вероватноћама јављати у осталим темама. Дакле, једна реч припада свакој од датих тема али са различитим вероватноћама.

Генерисање траженог документа могуће је извести следећом процедуром :

- Одабрати расподелу тема у документу - прецизирати која тема се са којим уделом појављује у документу. У конкретном примеру, расподела би могла бити : математика 30%, музика 30%, физика 20%, програмирање 15%, биологија 2% и хемија 3%.
- Докле год није достигнут тражени број речи
  - Изабрати тему из дистрибуције која је одабрана у 1.
  - Изабрати реч из теме која је одабрана у 2.а). Како свака тема има речи које фаворизује ( веће вероватноће у односу на остале речи ), то ће се те речи највероватније одабрати у овом кораку.



Описана процедура може се графички илустровати претходном сликом (8.16). Одабрана расподела тема у документу (тачка 1 описане процедуре) представљена је хистограмом на десној страни слике. Обојени кругови представљају одабир теме из документа (корак 2.а) ) док речи повезане стралицама са њима представљају одабирану реч из те теме (корак 2.б) ).<sup>1</sup>

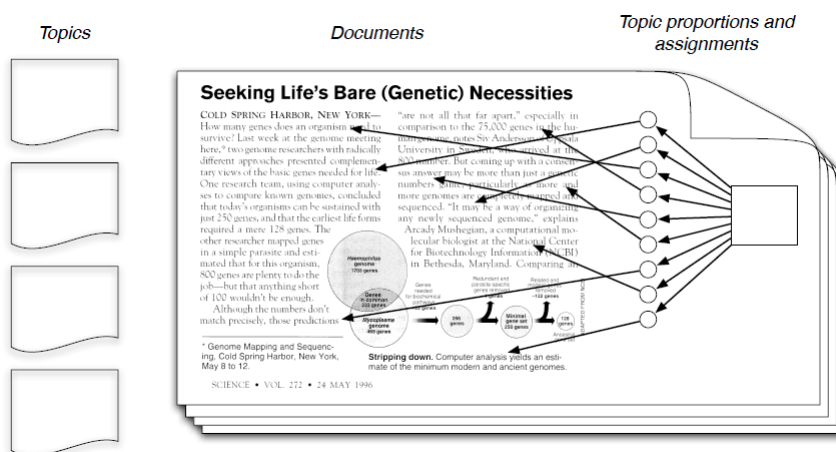
Формално, тема се дефинише као расподела речи над неким фиксним скупом речи - речником. Рецимо, тема биологија ће са већом вероватноћом садржати речи везане за ту област нпр. биљка, животиња, ћелија, ген итд. док ће тема везана за математику ове речи садржати са нижим вероватноћама у односу на речи нпр. број, разломак, променљива, коцка итд.

Дакле, основна карактеристика LDA-а је то што сви документи деле **исти скуп тема** али сваки документ те теме садржи у различитим односима. Овакаво посматрање докумената јако је природно и интуитивно.

### 2.1.2 Како се откривају теме - упрошћен пример

Проликом генерисања поменутог документа, било је познато која тема садржи које речи као и у којим односима је заступљена свака тема у тексту. Циљ алгоритама за моделовање тема је да **аутоматски** "открије" које су то теме присутне у неком документу и које речи припадају којој теми али **само** на основу речи које се јављају у документу, без било каквог додатног знања. Сазнање о томе која тема се у којој мери налази у неком документу, није од превеликог практичног значаја. Међутим, уколико је на располагању огромна количина докумената (нпр. дигитална база свих издања листа "Политика"), откривање сродних докумената, или докумената који се баве само одређеним темама може бити јако важно. Због тога ће се надаље говорити о скупу докумената над којим се извршава моделовање тема, уместо о једном документу. Притом, наравно, и даље важи претпоставка да сваки документ "говори о" свим темама које се могу издвојити из свих докумената, само у различитим односима.

Према свему реченом, једино што је *видљиво*, енг. **observed** су документи, односно речи које се у документима јављају. Тематска расподела по документима, као и расподела речи по темама су *скривене или невидљиве*, енг. *non observed, hidden* (Слика 2.2)



Слика 2.2: Пример чланка, преузето са [3]

Основни задатак алгорита је окривање скривених структура на основу видљивих. У овом тренутку, моделовање тема можемо посматрати као **обрнути** генеративни процес.

<sup>1</sup>Расподела на основу које се одабирају пропорције тема у кораку 1, назива се Дирихлеова расподела, енг. Dirichlet distribution. На основу те одабране расподеле, врши се придруживање речи документима, енг. allocate

Дакле, циљ моделовања тема је откривање скривених структура из којих су **највероватније**, генеративним процесом, добијени видљиве структуре, тј. документи. Током рада, откривају се удели различитих тема по документима као и расподеле речи унутар тема. Важно је напоменути да **именовање** тема не постоји у основној верзији алгоритама. Алгоритам групише речи у одређене целине - теме, а насловљавање тема се препушта стручњацима.

Нека је дат једноставан документ који, након склањања везника, личних заменица и осталих шумава (енг.*stopwords*) садржи речи приказане у следећој табели.

Etruscan	trade	price	temple	market

Слика 2.3: Преузето са [4]

Процес моделовања тема започео **случајним додељивањем** тема свакој од речи у документу. Дакле, пошто не постоји никакво знање о присуству тема у документима као ни о томе која реч припада којој теми, ову доделу је неопходно урадити на случајан начин. Интуитивно је јасно да се за тако нешто унапред мора одредити број тема који се захтева у задатом скупу докумената. Више о улазним параметрима алгорита може се наћи у одељку 3 овог рада. Пример једне случајне доделе дат је на следећој слици .

3	2	1	3	1
Etruscan	trade	price	temple	market

Слика 2.4: Преузето са [4]

На овај начин је направљена иницијална **расподела** тема унутар посматраног документа - 40% текста говори о теми 3, 40% о теми 1, док 20% говори о теми 2.

Уколико сада на сличан начин доделимо теме и осталим документима, полазни скуп докумената може се приказати следећом сликом.

trade	ship	2	3	2	1	1	Italy	temple	ship	trade	market
-------	------	---	---	---	---	---	-------	--------	------	-------	--------

Слика 2.5: Преузето са [4]

Пошто сваки документ има иницијалну, **случајну** расподелу тема, једноставно је груписати речи унутар тема и на тај начин направити иницијалну **случајну** расподелу речи по темама. Одређивање расподеле по темама, може се илустровати следећом сликом.

	1	2	3
Etruscan	1	0	35
market	50	0	1
price	42	1	0
temple	0	0	20
trade	10	8	1
...			

Слика 2.6: Преузето са [4]

У првој колони табеле уписане су све речи из свих докумената (речник) док је затим у свакој од наредних колона уписан одговарајући број који означава колико пута је дата реч додељена тој теми. Рецимо, бројеви 1 0 35 у другој врсти табеле означавају да је реч *Etruscan* 35 пута била сврстана у тему број 3, једном је била додељена теми број 1 док се ниједним није нашла у теми број 2.. Дакле, почевши од друге колоне приказане табеле, табела по колонама, садржи *расподелу речи по темама*. Једноставним сортирањем колона, добијају се највероватније речи у свакој од тема.

У овом тренутку, расподеле које су добијене нису релевантне зато што у позадини стоји апсолутно случајно додељивање тема које није базирано на документима, тј. на јединим видљивим подацима. Дакле, потребно је добијене резултате *прилагодити* тако да осликавају тематску структуру документа. Прилагођавање се одвија у одређеном, унапред познатом броју итерација. Генерално, што је већи број итерација, то су добијене расподеле релевантније, мада, како резултати показују, постоје и нека ограничења за ове вредности. Више о одређивању оптималног броја итерације може се наћи у одељцима 3 и 6 овог рада.


Унутар једне итерације, за сваки документ за сваку реч унутар тог документа врши се провера колико је тренутно додељена тема адекватна, тј. да ли постоји боља тема којој би та реч могла бити додељена. На тај начин, из итерације у итерацију, расподеле све више и више осликавају структуру докумената.

Нека је почетна расподела по темама дата на претходним сликама (2.5,2.6). Нека се провера подобности теме прво извршава за реч *trade* првог документа. Ова реч је унутар првог документа додељена теми број 2, док је, гледано са становишта свих докумената који се посматрају, укупно 8 пута сврстана у ту тему. Потребно је испитати да ли тема број 2 највише одговара тој речи. Претпоставимо да знамо теме за све остале речи, како из документа који се посматра, тако и за остале документе, и да је једино непознато којој теми припада *trade* у посматраном документу. Дакле, расподела тема у посматраном документу као и расподела речи по темама, сада изгледа као на следећој слици и потребно је доделити изабраној речи тему унутар посматраног документа.

3	?	1	3	1
Etruscan	trade	price	temple	market

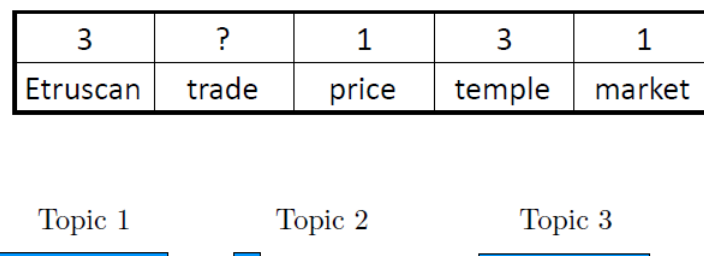
	1	2	3
Etruscan	1	0	35
market	50	0	1
price	42	1	0
temple	0	0	20
trade	10	7	1
...			



Слика 2.7: Преузето са [4]

Уколико се посматра расподела тема у изабраном документу, приметиће се да документ највише "говори о" темама 3 и 1 док о теми 2 не говори уопште. Према томе, удели тема 3 и 1 су значајни, док је удео теме 2 занемарљиво мали. Обзиром да је основна претпоставка овог модела да сви документи говоре о свим темама, не може се рећи да изабрани документ уопште "не говори" о теми 2. Начин на који ће се означити да је тема 2 јако слабо присутна у изабраном документу врши се тако што се теми 2 додели јако мали удео. Обзиром да је у питању реч из документа који има неки одређену расподелу тема, логично је очекивати да избор теме за ту реч зависи од тема тог документа.

Удели тема у изабраном документу могу се представити дужином линија, као што је приказано на следећој слици.



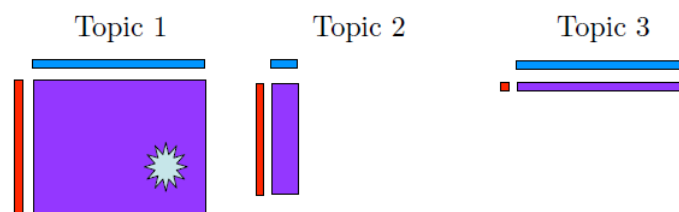
Слика 2.8: Преузето са [4]

Међутим, како је речник (скуп речи) заједнички за све документе, тема речи зависи и од глобалног присуства те речи у свим темама. И ова претпоставка је логична, јер, као што је на почетку наведено, свака реч припада свим темама, само са различитим вероватноћама. Глобално присуство изабране речи у свим темама приказано је на следећој слици :

	1	2	3
trade	10	7	1

Слика 2.9: Преузето са [4]

Дакле, избор теме за реч *trade* зависи од расподеле тема у посматраном документу као и од присуства те речи у свим темама. Ова зависност може се представити следећом сликом :



Слика 2.10: Преузето са [4]

Вертикална, црвена линија представља присуство речи у одговарајућој теми (формалније, вероватноћу са којом се та реч налази у изабраној теми). Љубичаста "површина" представља подобност да одговарајућа тема буде додељена тој речи у изабраном документу. Како се слике јасно може уочити, "највећу" површину формира тема 1 те је, према томе, речи *trade* додељује тема број 1 у овој итерацији. Након извршене измене, расподела тема у документу, као и расподела речи по темама, приказана је на следећој слици :

3	1	1	3	1
Etruscan	trade	price	temple	market

	1	2	3
Etruscan	1	0	35
market	50	0	1
price	42	1	0
temple	0	0	20
trade	11	7	1
...			

Слика 2.11: Преузето са [4]

На овај начин, изабраној речи додељена је највероватнија тема а тематска слика документа "више личи" на реалну слику.

Ако се описани процес примени на сваку реч сваког документа, расподеле ће из итерације у итерацију све више осликавати структуру полазних докумената. Обзиром да избор теме за сваку реч зависи од тренутно додљених расподела тема и речи, овим процесом се узимају у обзир видљиви подаци. Дакле, скривене структуре (непознате расподеле) се генеришу на основу једино видљивих података - докумената и речи у њима.

Узевши све претходно у обзир, јасно је због чега не постоји именовање тема. Када се процес "генерисања" заврши (нпр. достигне одређени број итерација), сортирањем одговарајућих колона из табеле са слике 2.7, добија се расподела речи по темама. На основу експертског знања овим расподелама додељују се имена - нпр. тема 1 - математика, тема 2- економија итд.

## Глава 3

# Математичка позадина

У претходним поглављима, рад се углавном бавио питањима *шта је ТМ алгоритам и чему служи*, без улажења у то **како** он уствари ради.

Опис рада ТМ алгоритама - конкретно LDA имплементације, биће изложен у неколико целина. Најпре ће се објаснити ( увести ) неки појмови вероватноће који су битни за разумевање суштине рада алгорита, а затим ће бити изнешена математичка позадина самог алгорита.

### 3.1 Теорија вероватноће

Теорија вероватноће је математичка дисциплина која се бави изучавањем случајних појава тј. појава чији исходи нису увек строго дефинисани.

Први проблеми који се могу сматрати проблемима вероватноће потичу још из 12. века и везани су за проучавање исхода разних игара на срећу. Развој *теорије вероватноће* почиње средином 17. века и везан је за имена Блеза Паскала, Пјера де Ферма и Кристијана Хајгенса. Наиме, између Паскала и Ферма је 1654. године започела интересантна преписка о низи проблема међу којима је био и проблем везан за поделу улога приликом прекида једне коцкарске игре. Проблем је био постављен на следећи начин : Два играча А и Б се договоре да читав улог припадне ономе ко први добије три игре. Када је играч А добио 2 игре а играч Б једну игру, играчи су споразумно одлучили да прекину игру. Поставља се питање како сада да поделе улог. Паскал је предложио поделу у односу 3:1 у корист играча А. Овај пример често се узима као почетак настанка теорије вероватноће.

Неке од појава које се догађају у реалном свету лако се могу предвидети и објаснити услед познавања законитости њиховог настанка. У такве појаве спадају нпр. помарачење Сунца и Месеца, плима и осека, гравитација итд. Међутим, постоје појаве чије узроке тренутно није могуће одредити па се не могу у потпуности објаснити и одредити. Неке од таквих појава су нпр. добитак на лутрији или метеоролошке појаве. Прилоком бацања металног, хомогеног новчића, никада није сигрно да ли ће пасти писмо или глава. Међутим, уколико бацамамо новчић много пута, може се уочити да је отприлике исти број пута пало писмо као и глава ( такве експерименте су радили Буфон и Пирсон Дакле, законитост код оваквих догађаја може се уочити тек након великог броја понављања појаве.

#### 3.1.1 Основни појмови

Основни полазни појам у теорији вероватноће је непразан скуп  $\Omega$  који представља скуп свих могућих исхода једног експеримента. Овај скуп се често назива **простор елементарних догађаја** и може бити коначан, пребројив или непребројив. **Случајни догађаја** или само **догађај** представља било који подскуп скупа  $\Omega$ . Најчешће се случајни догађаји означавају великим, штампаним, латиничним словима. За догађај **A** каже се да се **реализовао** ако

се реализовао неки исход  $\omega$  који припада скупу  $A$ . Догађај који је садржи све могуће елементарне исходе експеримента назива се **сигуран догађај** а догађај који не садржи ни један елементарни исход назива се **немогућ догађај**.

*Пример:* Нека је дата хомогена коцка чије су стране означене бројевима од 1 до 6. Елементарни догађаји су појављивање одређеног броја при бацању коцкице. Према томе, скуп свих могућих исхода експеримента бацања коцкице је  $\Omega = \{1, 2, 3, 4, 5, 6\}$ . Догађај  $A =$  "пао је паран број" одређује скуп  $A = \{2, 4, 6\}$

**Производ два догађаја** и, у ознаци  $AB$  је догађај који се реализује ако и само се ако реализују оба догађаја. Дакле, производ догађаја је пресек скупова  $A$  и  $B$ . Уколико су  $A$  и  $B$  дисјунктни скупови (пресек је празан скуп) за такве догађаје кажемо да су **несагласни** или да се **искључују**.

**Збир два догађаја**  $A$  и  $B$ , у ознаци  $A \cup B$  представља догађај који се реализује ако се реализује бар један од догађаја  $A$  и  $B$ .

**Разликом догађаја**  $A$  и  $B$ , у ознаци  $A - B$  назива се догађај који се реализује ако и само ако се реализује догађај  $A$  а не реализује догађај  $B$ .

**Потпун систем догађаја** : За догађаје  $A_1, A_2, \dots, A_n$  се каже да образују *потпун систем догађаја* уколико важи :  $\bigcup_i A_i = \Omega$  . Дакле, при реализацији неког експеримента бар један од догађаја  $A_1, A_2, \dots, A_n$  ће се реализовати. Посебно су интересантни потпуни системи несагласних догађаја као што се може видети код формуле тоталне вероватноће.

**Дефиниција 1 (Класична дефиниција вероватноће)** : Нека је  $\Omega = \{\omega_1, \omega_2, \dots, \omega_n\}$  скуп свих могућих једнаковероватних елементарних догађаја који су међусобно несагласни и нека је  $A = \{\omega_{i_1}, \omega_{i_2}, \dots, \omega_{i_m}\}$  догађај који се састоји од  $m$  елементарних једнаковероватних догађаја. Вероватноћа наступања догађаја  $A$  је :

$$P(A) = \frac{m}{n} \quad (3.1)$$

Претходна дефиниција може се неформално изразити и овако : вероватноћа догађаја  $A$  једнака је количнику броја **повољних исхода** експеримента ( исходи када се реализује догађај  $A$  ) и укупног броја свих могућих исхода експеримента.

Класична дефиниција вероватноће је применљива само онде где су елементарни догађаји једнаковероватни. Међутим, тај услов је у пракси јако тешко испунити. Чак и у случајевима када је то наизглед очигледно, као што је бацање коцкице, једнаковероватност не може бити гарантована. Разлози за то могу бити технологија израде коцкице која не мора бити савршено прецизна, немогућност обезбеђивања идеалних и непромењљивих услова током извођења експеримента итд. Због тога је једини начин којим је могуће заиста утврдити вероватноћу догађаја  $A$  *статистички приступ* заснован на великом броју експеримената.

**Дефиниција 2 (Статистичка дефиниција вероватноће)** : Нека се у  $n$  понављања експеримента изведених под приближно истим условима догађај  $A$  реализовао  $m_n$  пута. Вероватноћа догађаја  $A$  је

$$P(A) = \lim_{n \rightarrow \infty} \frac{m_n}{n} \quad (3.2)$$

### 3.1.2 Условна вероватноћа

Вероватноћа догађаја чија реализација **не зависи** од наступања било ког другог догађаја назива се **безусловна вероватноћа**. Ако је реализација догађаја  $A$  условљена реализацијом неког догађаја  $B$  при чему  $B$  није немогућ догађај (  $P(B) \neq 0$  ), тада се вероватноћа догађаја под условом да се десио догађај  $B$  назива **условном вероватноћом** и означава се са  $P(A | B)$ . Дакле,  $P(A | B)$  је вероватноћа догађаја  $A$  под условима који сигурно доводе до реализације догађаја  $B$ .

Нека се изводи експеримент у коме постоји  $n$  једнаковероватних елементарних догађаја и нека је са  $n_A, n_B, n_{AB}$  означен број елементарних догађаја који доводе до реализације догађаја  $A, B, AB$  редом.

Према класичној дефиницији вероватноће, вероватноћа реализације догађаја  $A$  и  $AB$  је :

$$P(B) = \frac{n_B}{n}, P(AB) = \frac{n_{AB}}{n} \quad (3.3)$$

Ако је реализација догађаја  $A$  условљена реализацијом догађаја  $B$ , то је број повољних исхода догађаја  $A$   $n_{AB}$  (број елементарних догађаја који имају осбине и скупа  $A$  и скупа  $B$ ). Пошто се догађај  $A$  реализује само ако се реализовао догађај  $B$ , број свих могућих исхода је  $n_B$  (број свих могућих елементарних догађаја када наступа догађај  $B$ ). Дакле, условна вероватноћа догађаја  $A$ , под условом да се десио догађај  $B$  је :

$$P(A | B) = \frac{n_{AB}}{n_B} = \frac{\frac{n_{AB}}{n}}{\frac{n_B}{n}} = \frac{P(AB)}{P(B)}, P(B) \neq 0 \quad (3.4)$$

У случају да је догађај  $B$  условљен догађајом  $A$ , аналогно се изводи да је

$$P(B | A) = \frac{P(AB)}{P(A)}, P(A) \neq 0 \quad (3.5)$$

Из релација (3.4) и (3.5) следи

$$P(AB) = P(B) \cdot P(A | B) = P(A) \cdot P(B | A) \quad (3.6)$$

Релација (3.6) назива се још и **теорема о производу вероватноћа**

**Теорема 1 (Формула тоталне вероватноће)** : Ако су  $H_1, H_2, \dots, H_n$  међусобно несагласни догађаји,  $P(H_i) > 0 (i = 1, \dots, n)$  при чему важи  $H_1 + H_2 + \dots + H_n = \Omega$  тада је :

$$P(A) = \sum_{i=1}^n P(H_i)P(A | H_i) \text{ за сваки догађај } A \subseteq \Omega \quad (3.7)$$

Напомена : Догађаји  $H_1, H_2, \dots, H_n$  чине потпун систем несагласних догађаја.

**Доказ 1** Обзором да су догађаји подскупови скупа свих елементарних догађаја очигледно је да важи

$$A = A\Omega = A \sum_{i=1}^n H_i = \sum_{i=1}^n AH_i. \quad (3.8)$$

На основу релације (3.6) следи :

$$P(A) = P\left(\sum_{i=1}^n AH_i\right) = \sum_{i=1}^n P(AH_i) = \sum_{i=1}^n P(H_i)P(A | H_i) \quad (3.9)$$

Вероватноће  $P(H_i)$  су обично познате унапред и називају се **априорним вероватноћама** а сами догађаји **хипотезама**.

**Теорема 2 (Бајесова формула <sup>1</sup>)** : Ако су  $H_1, H_2, \dots, H_n$  међусобно несагласни догађаји,  $P(H_i) > 0 (i = 1, \dots, n)$  при чему важи  $H_1 + H_2 + \dots + H_n = \Omega$  тада је :

$$P(H_i | A) = \frac{P(H_i)P(A | H_i)}{\sum_{j=1}^n P(H_j)P(A | H_j)} \quad (i = 1 \dots n) \quad \text{за сваки догађај } A \subseteq \Omega \quad (3.10)$$



**Доказ 2** Из релације (3.6) следи :

$$P(H_i A) = P(H_i)P(A | H_i) = P(A)P(H_i | A) \quad (i = 1...n) \quad (3.11)$$

Условна вероватноћа догађаја  $H_i$  под условом да се десио догађај  $A$  је:

$$P(H_i | A) = \frac{P(H_i A)}{P(A)} = \frac{P(H_i)P(A | H_i)}{P(A)}$$

Примењујући формулу потпуне вероватноће за  $P(A)$  добија се

$$P(H_i | A) = \frac{P(H_i)P(A | H_i)}{\sum_{j=1}^n P(H_j)P(A | H_j)}$$

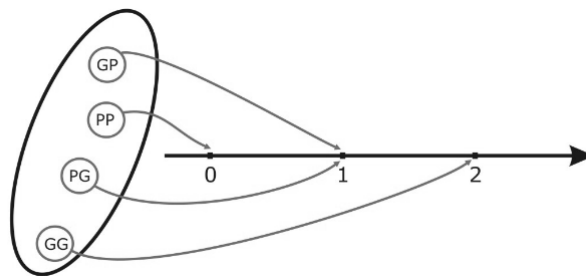
што представља Бајесову формулу.

### 3.1.3 Случајне променљиве

Ако се сваком елементарном догађају придружи један реалан број, онда се извођење експеримента може посматрати као избор вредности једне променљиве. Променљива величина која те бројене вредности узима са одређеним вероватноћама назива се *случајна променљива*. Дакле, уместо вербалне карактеризације догађаја ( описа речима шта догађај представља ) много је једноставније за рад догађаје окарактерисати бројним вредностима тј. неким реалним бројевима.

**Пример 1 :** У експерименту бацања новчића могућа су два елементарна исхода : грб или писмо. Нека је догађај који се посматра "пало је писмо". Појава писма се може означити бројем 1 а појава грба бројем 0. Сада се овај експеримент може замислити као избор 0 или 1 са вероватноћом  $\frac{1}{2}$

**Пример 2 :** Новчић се баца два пута. Нека је са  $P$  означена појава писма а са  $G$  појава грба. Скуп свих елементарних исхода експеримента је  $\Omega = \{PP, PG, GP, GG\}$ . Нека је догађај који се посматра "број палих писама". Сваком исходу се може доделити један реалан број и то  $PP \rightarrow 2, GP \rightarrow 1, PG \rightarrow 1, GG \rightarrow 0$ . Ово додељивање вредности се карактерише случајном променљивом. Случајна променљива сваку од ових вредности узима са различитом вероватноћом.



Слика 3.1: Графички пример случајне променљиве

**Дефиниција 3** Функција  $X$  која сваком случајном догађају  $\omega \in \Omega$  додељује неки реалан број  $X(\omega)$  назива се *случајна променљива* где је  $X : \Omega \rightarrow R$

Дакле, случајна променљива је пресликавање скупа  $\Omega$  у скуп **реалних** бројева за разлику од вероватноће која је пресликавање скупа  $\Omega$  у скуп  $[0, 1]$

Важно је уочити да случајна променљива **нема конкретну вредност** већ се само говори о вероватноћама да узме неки конкретну вредност.

Разликују се два основна типа случајних променљивих - **дискретне** и **непрекидне**. Подела се врши у зависности од тога да ли случајна променљива узима вредности у пребројивом или непребројивом скупу вредности.

### 3.1.3.1 Дискретне случајне променљиве

За случајну променљиву се каже да је дискретног типа ако узима коначан број изолованих вредности или пребројиво много вредности

**Дефиниција 4** Нека случајна променљива  $X$  може да узме вредности  $x_1, x_2, \dots, x_n$  са вероватноћама  $p_1, p_2, \dots, p_n$  при чему важи да је  $p_1 + p_2 + \dots + p_n = 1$ . Скуп парова  $(x_i, p_i = P\{X = x_i\})$ ,  $i = 1, 2, \dots, n$  или написано :

$$\begin{pmatrix} x_1 & x_2 & \cdots \\ p(x_1) & p(x_2) & \cdots \end{pmatrix}$$

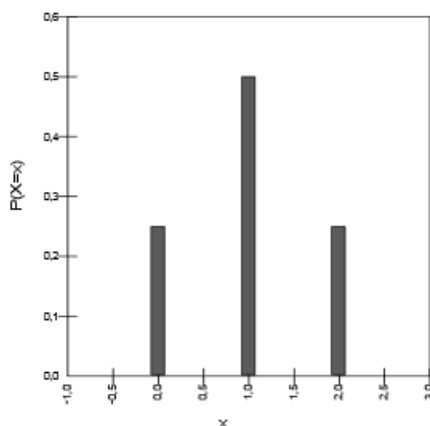
чине закон расподеле или распоред вероватноћа случајне променљиве  $X$ .

Закон расподеле случајне променљиве може да се посматра као **правило** по коме се свакој вредности случајне променљиве придружује одговарајућа вероватноћа. Дакле, при реализацији експеримента сигурно ће се десити догађај којем је придружена нека вредност случајне променљиве. Због тога је сума свих вероватноћа у расподели случајне променљиве 1. Међутим, нису све вредности подједнако вероватне па се свакој вредности придружује вероватноћа са којом се очекује. Претходна дефиниција може се интерпретирати и на следећи начин : извесна маса једнака јединици је распоређена на такав начин да се у тачкама  $x_1, x_2, \dots, x_n$  налазе одговарајући делови масе  $p_1, p_2, \dots, p_n$ . Услед оваквог тумачења, закон расподеле вероватноћа се често назива и **функција масе вероватноћа**

У примеру 2, случајна променљива може да узме три вредности, тј. писмо се може појавити 0, 1 или 2 пута у два бацања. Ни један други исход није могућ - нпр. у два бацања писмо не може да се појави 3 пута или -1 пут. Међутим, вероватноћа да се писмо неће појавити ни једном (или да се појави два пута) је  $\frac{1}{4}$  - вероватноћа да падне глава у првом бацању је  $\frac{1}{2}$  и вероватноћа да падне глава у другом бацању је  $\frac{1}{2}$ , дакле, вероватноћа да оба пута падне глава је  $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ , вероватноћа да се писмо појави тачно једном је  $\frac{1}{2}$  - писмо пада тачно једном у случају PG или GP. Вероватноћа за оба ова догађаја је  $\frac{1}{4}$ . Дакле, вероватноћа да се десио бар један од ових догађаја је  $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ . Према томе, расподела случајне променљиве "број појављивања писма у два бацања" је :

$$\begin{pmatrix} 0 & 1 & 2 \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{pmatrix}$$

Закон расподеле дискретне случајне променљиве може се представити графички, као на следећој слици :



Слика 3.2: Графички пример случајне променљиве

На апсциси се налазе могуће вредности случајних променљивих док се на ординати налазе вероватноће са којом случајна променљива узима дату вредности. Са претходне слике јасно се уочава дискретност посматране случајне променљиве - вероватноћа да случајна променљива узме вредност између неке две целобројне вредности је 0.

### Функција расподеле дискретне случајне променљиве:

Распоред или закон расподеле случајне променљиве дискретног типа може се представити као листа свих могућих вредности случајне променљиве и одговарајућих вероватноћа. Међутим, поставља се питање како представити случајну променљиву која може узимати јако пуно вредности тј. бесконачно много вредности. У овом случају би требало формирати листу од бесконачно много чланова, што је практично неизводљиво. (Пример једне такве случајне променљиве би био - број бацања коцкице док се не добију две узастопне шестике. Случајна променљива може узети вредности 2,3,4,... са различитим вероватноћама, при чему не постоји горња граница броја бацања при којој се сигурно добијају две узастопне шестике). Због описаног проблема, потребно је пронаћи другачији начин представљања случајне променљиве и одговарајућих вероватноћа. То се постиже **функцијом расподеле** која се може дефинисати за сваку случајну променљиву.

**Дефиниција 5 Функција расподеле** ( још се назива и **кумулативна функција расподеле**) дискретне случајне променљиве претставља вероватноћу да случајна променљива  $X$  узме вредност која је мања или једнака неком реалном броју  $x$  при чему је дефинисана за свако реално  $x$ .

$$F(x) = P(X \leq x) \quad \forall x \in R$$

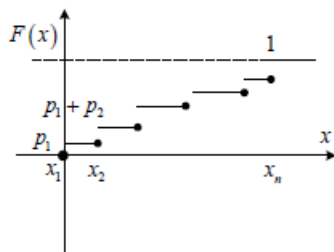
Дакле, кумулативна функција расподеле има облик

$$F(x) = \begin{cases} 0, & x \leq x_1 \\ p_1, & x_1 < x \leq x_2 \\ p_1 + p_2, & x_2 < x \leq x_3 \\ \dots & \dots \\ 1 & x > x_n \end{cases}$$

и може се изразити као :

$$F(x) = \sum_{k, x_k \leq x} P(X = x_k)$$

а графички приказ је дат на следећој слици :



Слика 3.3: График кумулативне функције расподеле случајне променљиве дискретног типа

Две најважније дискретне расподеле су **Биномна** и **Пуасонова** расподела.

### 3.1.3.2 Непрекидне случајне променљиве

Случајна променљива је (апсолутно) непрекидног типа ако може да узме **било коју** вредност из неког интервала. Број вредности које може да узме случајна променљива непрекидног типа је **бесконачан**. Неки од примера су : висина и тежина људи, дужина трајања батерије итд. На пример, нека је  $X$  случајна променљива која представља дужину рада сијалице. Ова случајна променљива може узети било коју вредност на интервалу од 1 до нпр. 1000 сати. Како у интервалу  $[0, 1000]$  има бесконачно много реалних бројева, не постоји начин да се дефинише вероватноћа за сваку појединачну вредност, као што је био случај код дискретних променљивих. Такође, интуитивно је јасно да је вероватноћа да ће сијалица прегорети у тачно одређеном тренутку једнака 0 док је вероватноћа да ће прегорети у неком временском интервалу различита од нуле.

**Дефиниција 6** Случајна променљива  $X$  је апсолутно непрекидног типа ако постоји **ненегативна** функција  $f: \mathbb{R} \rightarrow \mathbb{R}$  таква да за било који интервал  $[a, b] \subset (-\infty, \infty)$  важи :

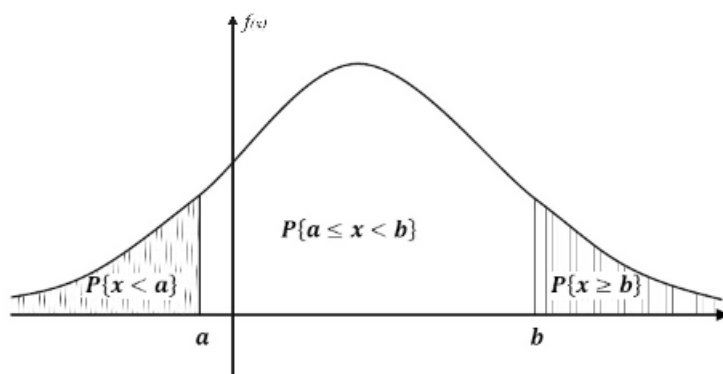
$$P\{a \leq X < b\} = \int_a^b f(x)dx \quad (3.12)$$

Функција  $f(x)$  мора да задовољи услов :

$$P\{-\infty \leq X < \infty\} = P\{\Omega\} = \int_{-\infty}^{\infty} f(x)dx = 1$$

Функција  $f(x)$  се назива **густина расподеле вероватноће** случајне променљиве  $X$ . Случајне променљиве **дискретног типа** немају густину расподеле баш као што ни случајне променљиве непрекидног типа немају закон расподеле вероватноћа.

Из релације (3.12) следи да је вероватноћа да случајна променљива узме вредност из интервала  $[a, b]$  једнака **површини** испод графика функције  $f(x)$  на интервалу  $[a, b]$ .



Слика 3.4: Функција густине

**Функција расподеле непрекидне случајне променљиве:**

**Дефиниција 7** *Функција расподеле (кумулативна функција расподеле) непрекидне случајне променљиве се може представити као :*

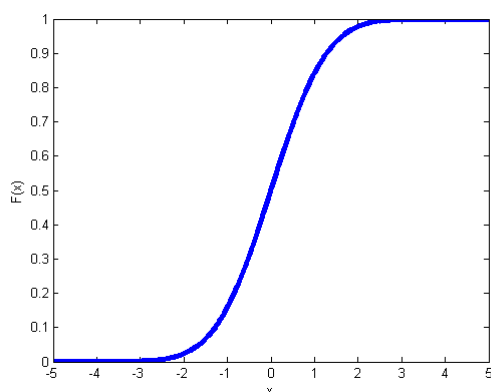
$$F(x) = P(X \leq x) = \int_{-\infty}^x f(t)dt \quad x \in (-\infty, \infty)$$

где је  $f(x)$  функција густине.

Дефиниција кумулативне суме преко интеграла је јаснија ако се има на уму интервал из ког случајна променљива може да узме вредности. Код случајних променљивих дискретног типа, тај скуп је био пребројив па се кумулативна функција расподеле дефинисала преко суме. Случајне променљиве непрекидног типа могу узети бесконачно много вредности па се сума код дискретних случајних променљивих ( када број тачака тежи у бесконачност) замењује интегралом.

Напомена : Ако случајна променљива  $X$  не узима све вредности из интервала  $(-\infty, \infty)$  усваја се да је  $f(x) = 0$  за све вредности  $x$  из интервала у којима  $X$  не узима вредности.

График кумулативне функције расподеле непрекидне случајне променљиве  $X$  је сада представљен глатком кривом линијом ( за разлику од случајне променљиве дискретног типа где је график био "степенаст").



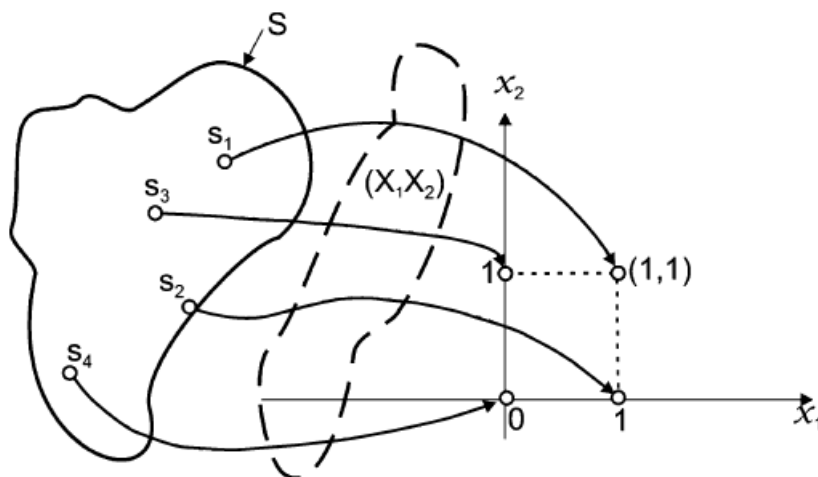
Слика 3.5: Кумулативна функција расподеле за случајне променљиве непрекидног типа

### 3.1.3.3 Вишедимензионалне случајне променљиве

Случајна променљива представља пресликавање скупа догађаја у реалне бројеве. Дакле, излази експеримента се мапирају у једнодимензионалан простор реалних бројева. Међутим, постоје случајеви када је потребно излазе експеримента мапирати у вишедимензионалне реалне просторе. На пример, при истовременом бацању два новчића могућа су 4 исхода :

1.  $s_1$ : први новчић писмо - други новчић писмо
2.  $s_2$ : први новчић писмо - други новчић глава
3.  $s_3$ : први новчић глава - други новчић писмо
4.  $s_4$ : први новчић глава - други новчић глава

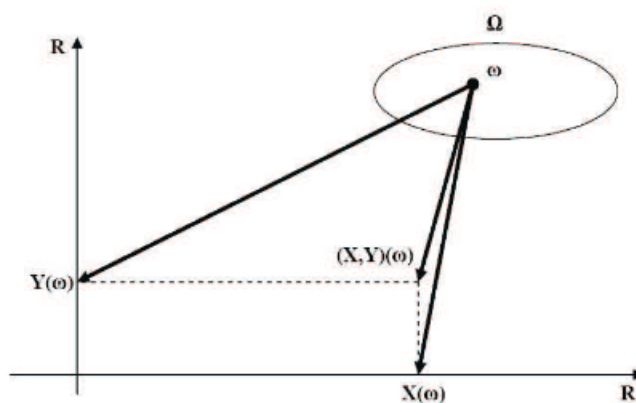
Нека је са  $X_1$  означена случајна променљива која узима вредност 1 ако се на првом новчићу појавила глава, односно 0 ако се појавило писмо и аналогно  $X_2$  која на исти начин означава појаву главе на другом новчићу. Исход експеримента се сада може описати дводимензионалном променљивом  $(X_1, X_2)$ . Графички приказ ове дводимензионалне променљиве дат је на следећој слици :



Слика 3.6: Експеримент : бацање два новчића.  
Скуп  $S$  представља скуп свих елементарних исхода ( $\Omega$ )

**Дефиниција 8** Ако су  $X : \Omega \rightarrow \mathbb{R}, Y : \Omega \rightarrow \mathbb{R}$  случајне променљиве, тада се **уређени пар**  $(X, Y)$  назива **двостепенациона случајна променљива**. Уређеним паром  $(X, Y)$  се сваком исходу  $\omega \in \Omega$  придружује уређени пар бројева  $(X(\omega), Y(\omega)) = (x, y) \in \mathbb{R} \times \mathbb{R} = \mathbb{R}^2$ .

На следећој слици графички је представљена двостепенациона случајна променљива.



Слика 3.7: Двостепенациона случајна променљива

Овако уведен појам двостепенационале случајне променљиве се може проширити и на више димензија и тада настају  $n$ -димензионалне случајне променљиве. Закључци изведени за двостепенационалне се такође односе и на вишестепенационалне случајне променљиве.

**Кумулативна функција расподеле** ( још се назива и *заједничка расподела* енгл. *joint distribution* двостепенационале случајне променљиве, у ознаци  $F_{XY} : \mathbb{R}^2 \rightarrow [0, 1]$  дефинише се као вероватноћа реализације догађаја  $\{X \leq x, Y \leq y\}$  односно :

$$F_{X,Y}(x, y) = P\{X \leq x, Y \leq y\} \quad -\infty < x, y < \infty$$

Неке карактеристике функције расподеле двостепенационале случајне променљиве :

1.  $0 \leq F_{X_1, X_2}(x_1, x_2) \leq 1$
2.  $F_{X_1, X_2}(-\infty, -\infty) = 0$

$$3. F_{X_1, X_2}(-\infty, -\infty) = 0$$

$$F_{X_1, X_2}(-\infty, x_2) = 0 \quad F_{X_1, X_2}(x_1, -\infty) = 0$$

$$4. F_{X_1, X_2}(\infty, \infty) = 1$$

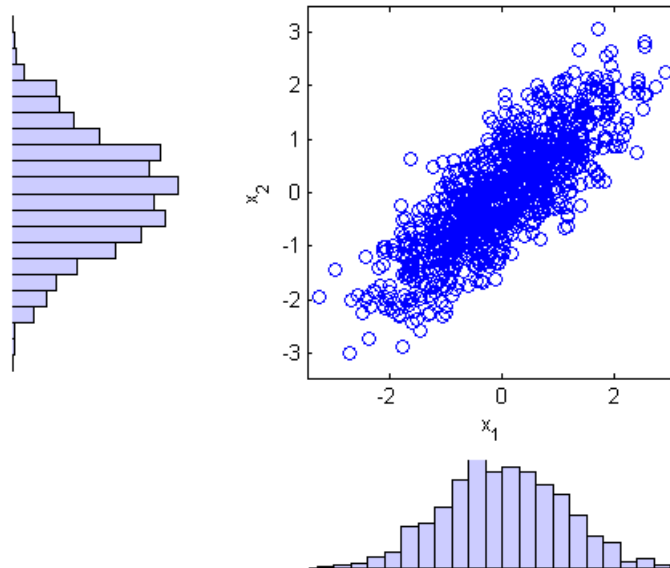
5.

$$F_{X_1, X_2}(x_1, \infty) = F_{X_1}(x_1) \quad (3.13)$$

6.

$$F_{X_1, X_2}(\infty, x_2) = F_{X_2}(x_2) \quad (3.14)$$

Једнакостима (3.13) и (3.14) су дефинисане **маргиналне расподеле** случајних променљивих  $X_1$  и  $X_2$ . Маргиналне расподеле су уствари расподеле једнодиментионалних случајних променљивих  $X_1$  и  $X_2$ . На следећој слици је представљена заједничка расподела две случајне променљиве заједно са њиховим маргиналним расподелама.

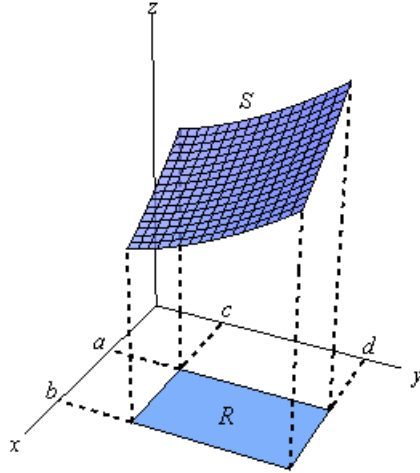


Слика 3.8: Дводимензионална случајна променљива и маргиналне расподеле

**Заједничка функција густине** (енг. joint density function) дводимензионалне случајне променљиве се дефинише као :

$$f_{X_1, X_2}(x_1, x_2) = \frac{d^2 F_{X_1, X_2}(x_1, x_2)}{dx_1 dx_2} \quad (3.15)$$

У случају једнодимензионалне случајне променљиве, површина испод графика функције густине на неком интервалу представља је вероватноћу да случајна променљива узме вредност из тог интервала. У случају дводимензионалне случајне променљиве од интереса је пронаћи вероватноћу да она узме вредност из неке **области**. Та вероватноћа представља **запремину** тела ограниченог функцијом густине са горње стране и датом облашћу са доње стране.



Слика 3.9: Вероватноћа да дводемпзионална случајна промељива  $(X, Y)$  узме вредности из области  $R$

Неке особине функције густине :

1.  $\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X_1, X_2}(x_1, x_2) dx_1 dx_2 = 1$
2.  $F_{X_1, X_2}(x_1, x_2) = \int_{-\infty}^{x_1} \int_{-\infty}^{x_2} f_{X_1, X_2}(x_1, x_2) dx_1 dx_2$
3.  $F_{X_1}(x_1) = \int_{-\infty}^{x_1} \int_{-\infty}^{\infty} f_{X_1, X_2}(x_1, x_2) dx_1 dx_2$   
 $F_{X_2}(x_2) = \int_{-\infty}^{x_2} \int_{-\infty}^{\infty} f_{X_1, X_2}(x_1, x_2) dx_1 dx_2$
4.  $f_{X_1}(x_1) = \int_{-\infty}^{\infty} f_{X_1, X_2}(x_1, x_2) dx_2$   
 $f_{X_2}(x_2) = \int_{-\infty}^{\infty} f_{X_1, X_2}(x_1, x_2) dx_1$
5.  $P\{x_{11} < X_1 \leq x_{12}, x_{21} < X_2 \leq x_{22}\} = \int_{x_{21}}^{x_{22}} \int_{x_{11}}^{x_{12}} f_{X_1, X_2}(x_1, x_2) dx_1 dx_2$

### Функција условне расподеле и густине

У неким специфичним случајевима је потребно пронаћи расподелу једне случајне променљиве знајући вредност друге случајне променљиве. Таква расподела назива се **условном расподелом** и обележава се са  $F_{X_1}(x_1 | X_2 = x_2)$ . Аналогно, може се дефинисати и проблем налажења функције густине једне случајне променљиве знајући вредност друге случајне променљиве и таква функција густине се означава са  $f_{X_1}(x_1 | X_2 = x_2)$ . Према [8] условна расподела односно густина се рачуна по следећем обрасцу ( детаљно извођење се може наћи у [8])

$$F_{X_1}(x_1 | X_2 = x_2) = \frac{\int_{-\infty}^{x_1} f_{X_1, X_2}(x_1, x_2) dx_1}{f_{X_2}(x_2)}$$

односно :

$$f_{X_1}(x_1 | X_2 = x_2) = \frac{f_{X_1, X_2}(x_1, x_2)}{f_{X_2}(x_2)}$$



## 3.2 Важније расподеле

### 3.2.1 Биномна и полиномна(енг. multinomial ) расподела

#### Биномна расподела

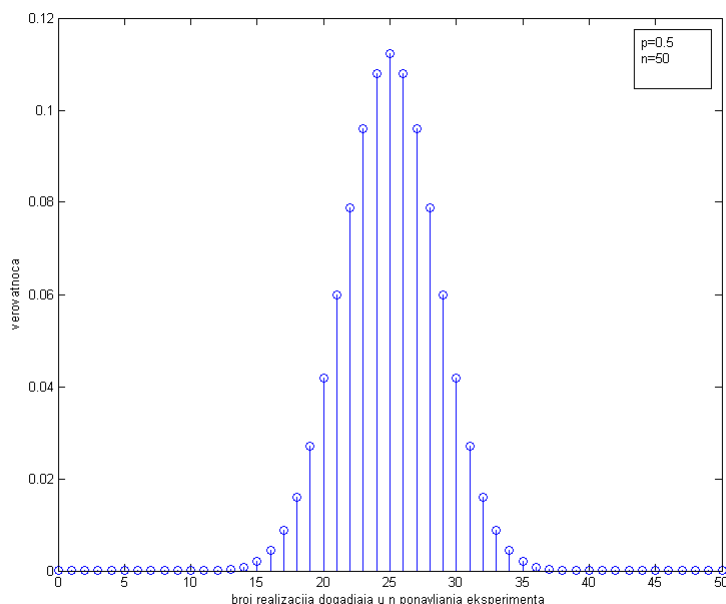
Нека ја  $A$  догађај неког експеримента  $E$  који се реализује са вероватноћом  $P(A) = p$ . Тада је вероватноћа супротног догађаја  $P(\bar{A}) = 1 - p = q$ . Резултат експеримента који је до интереса је остваривање или неостваривање догађаја  $A$ . Дакле, може се сматрати да је скуп свих елементарних исхода  $\Omega = \{A, \bar{A}\}$ . Нека се експеримент понавља **независно** и у неизмењеним условима  $n$  пута. На тај начин је формиран **сложени експеримент** чији скуп елементарних исхода садржи све могуће  $n$ -торке састављене од  $A$  и  $\bar{A}$  и има их укупно  $2^n$ . Нека је, даље, на том скупу елементарних исхода дефинисана случајна променљива  $X_n$  као број остваривања догађаја  $A$  у  $n$  понављања експеримента  $E$ . Вероватноћа да ова случајна променљива узме конкретну вредност  $k$  је :

$$p_k = P\{X_n = k\} = \binom{n}{k} p^k q^{n-k}$$

Вероватноће  $P\{X_n = k\}, (k = 0, 1, \dots, n)$  дефинишу **биномну расподелу**, у ознаци  $\mathbb{B}(n, p)$ . Ова расподела је дискретног типа а њена функција расподеле(кумулятивна) се може изразити као :

$$F(x) = \begin{cases} 0 & , x \leq 0 \\ \sum_{k=0}^r \binom{n}{k} p^k q^{n-k} & 0 < r < n \\ 1 & x > n \end{cases}$$

Закон расподеле вероватноћа случајне променљиве биномне расподеле приказан је на следећој слици :



Слика 3.10: Биномна расподела - закон расподеле

#### Полиномна (енг. multivariate) расподела

Изводи се серија од  $n$  независних експеримената при чему резултат експеримента може бити један од **коначно много** догађаја :  $A_1, A_2, \dots, A_k, \sum_{i=1}^k A_i = \Omega, P(A_i) = p_i (i = 1, 2, \dots, k)$ . Ако се дефинише  $k$ -диментионална случајна применљива  $(S_n^{(1)}, \dots, S_n^{(k)})$ , где  $S_n^{(i)}$  представља број релаизација случајног догађаја  $A_i$  у  $n$  независних експеримената, тада важи :

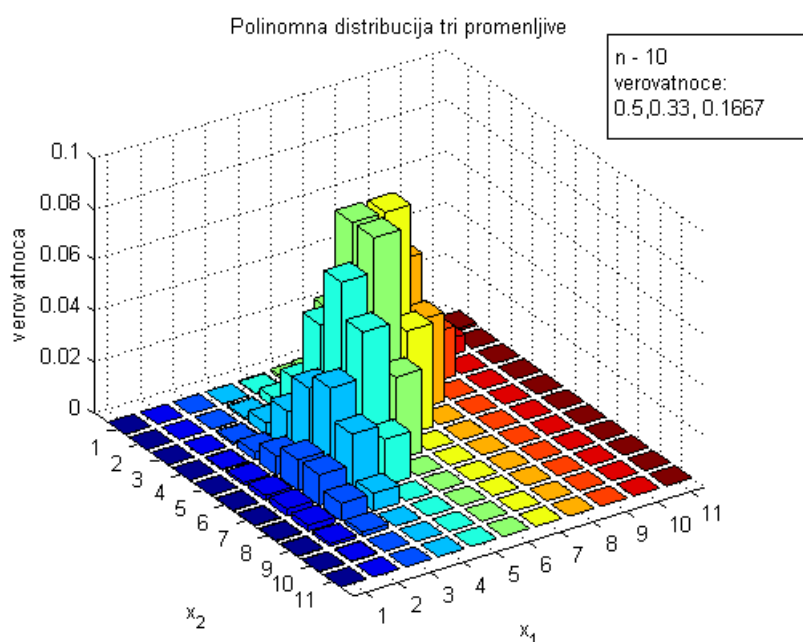
$$P(S_n^{(1)} = r_1, \dots, S_n^{(k)} = r_k) = \frac{n!}{r_1! \dots r_k!} p_1^{r_1} \dots p_k^{r_k}$$

$$r_1, \dots, r_k \in \{0, 1, \dots, n\} \quad r_1 + \dots + r_k = n$$

Ако се са  $S = (S_n^{(1)}, \dots, S_n^{(k)})$  означи  $k$ -диментионална случајна применљива која има полиномијалну расподелу тада се то записује као :

$$S \sim Mult(n, p)$$

где је  $p = (p_1, p_2, \dots, p_k)$  Пример полиномне расподеле при чему резултат експеримента може бити један од **три** догађаја, дат је на следећој слици :



Слика 3.11: Биномна расподела - закон расподеле

### 3.2.2 Дирихлеова расподела

Дирихлеова расподела представља фамилију расподела за параметре  $p$  полиномијалне расподеле. Задаје се са :

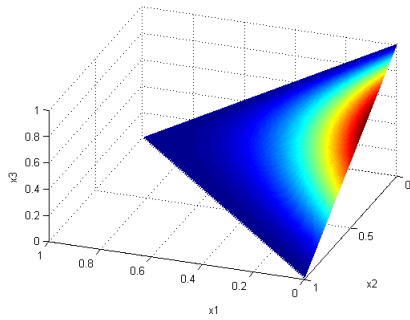
$$Dir(p; \alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^K p_k^{\alpha_k - 1}$$

при чему је  $\alpha$  параметар расподеле а  $B$  означава мултиномијалну бета функцију. Мултиномијалну бета функција се изражава преко гама функције на следећи начин

$$B(\alpha) = \frac{\prod_{i=1}^{|\alpha|} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{|\alpha|} \alpha_i)}$$

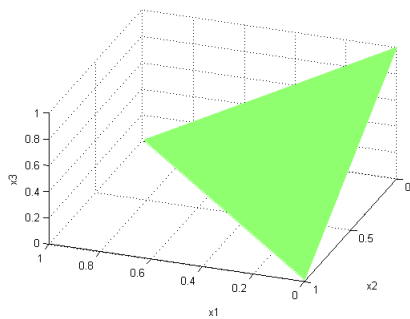
На слећој слици је графички представљена Дирихлеова расподела за три променљиве :

- $\alpha = (1, 2, 3)$



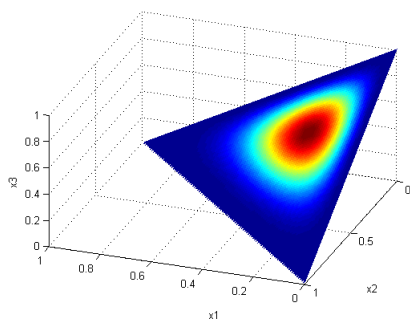
Слика 3.12: Дирихлеова расподела - интензивнија боја предтсваља већу вероватноћу

- $\alpha = (1, 1, 1)$

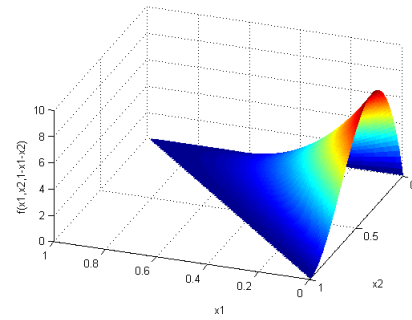


Слика 3.14: Дирихлеова расподела - интензивнија боја предтсваља већу вероватноћу

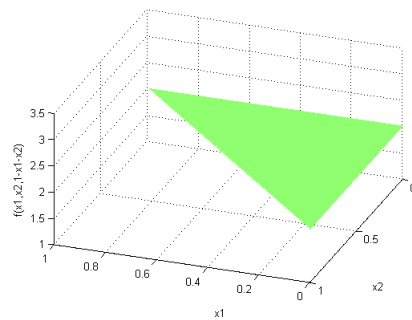
- $\alpha = (3, 3, 5)$



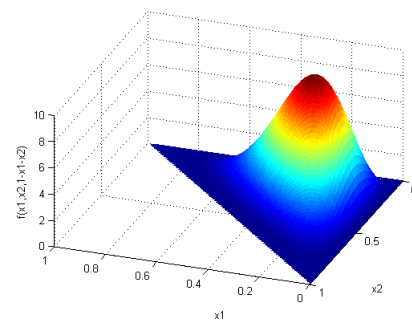
Слика 3.16: Дирихлеова расподела - интензивнија боја предтсваља већу вероватноћу



Слика 3.13: Дирихлеова расподела у три димензије



Слика 3.15: Дирихлеова расподела у три димензије



Слика 3.17: Дирихлеова расподела у три димензије

### 3.3 Гибсово узорковање

#### 3.3.1 Марковљеви ланци

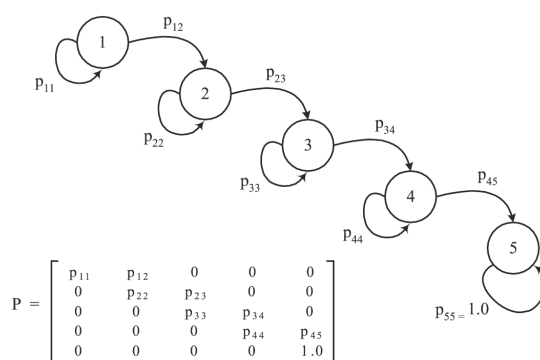
Марковљевим ланцима моделује се математички сиситем стања и прелаза међу тим стањима.

**Дефиниција 9** *Случајан (стохастички) процес представља математички модел процеса чија је еволуција описана законима вероватноће.*

*Марковљеви процеси су они случајни процеси чије будуће стање зависи само од тренутног стања. Оваква особина још се назива и одсуство памћења*

*Марковљеви ланци представљају посебну врсту Марковљевих процеса где се процес може налазити само у коначном броју стања.*

Пример Марковљевог ланца дат је на следећој слици :



Слика 3.18: Марковљев ланац - графички пример и матрица транзиције

Систем се састоји од 5 стања. У сваком стању, са одређеним вероватноћама систем може да пређе у неко од следећих стања - конкретно да остане у тренутном стању или да пређе у једно стање ниже. Вероватноћа преласка у следеће стање зависи само од тренутног стања. Марковљеви ланци се често представљају **матрицама транзиције** при чему  $i$ -та врста у матрици садржи вероватноће преласка у свако од стање система када се систем налази у стању  $i$ . Сума свих вероватноћа у свакој врсти је 1 ( систем сигурно мора да се нађе у неком стању, дакле вероватноћа да систем пређе у неко стање, могуће исто, је 1 ). Свака врста представља условни закон расподеле вероватноћа да систем пређе у било које стање у односу на тренутно стање (  $i$ -та врста -  $i$ -то стање ). Сака колона представља маргиналну расподелу вероватноћа да се систем нађе у одређеном стању (  $i$ -та колона -  $i$ -то стање ).

Систем се у једном тренутку може налазити у само једном стању. Нека се стање система карактерише случајном променљивом  $X_n$  која у тренутку  $n$  има расподелу  $\vec{s}$  и нека је укупан број стања система  $M$ . Расподела  $\vec{s}$  је у ствари закон расподеле (енг. PMF) јер се ради о случајној променљивој дискретног типа - систем може бити само у једном од  $M$  могућих стања, и конкретно може се посматрати као вектор врсте, димензија  $1 \times M$  где се на  $i$  - том месту налази вероватноћа да се систем у тренутку  $n$  нађе у стању  $i$ .

У следећем временском тренутку,  $n + 1$ , систем се може наћи у било ком од  $M$  стања са различитим вероватноћама. Вероватноћа да ће се систем у тренутку  $n + 1$  наћи у стању  $j$  означава се са  $P(X_{n+1} = j)$ . Пошто ова вероватноћа зависи од стања у претходном тренутку, може се изразити на следећи начин ( према формули тоталне вероватноће )

$$P(X_{n+1} = j) = \sum_i^M P(X_{n+1} = j \mid X_n = i)P(X_n = i) = *$$

$P(X_{n+1} = j \mid X_n = i)$  = вероватноћа преласка система из стања  $i$  у стање  $j \rightarrow p_{i,j}$   
 $P(X_n = i)$  = вероватноћа да се систем у тренутку  $n$  нађе у стању  $i \rightarrow s_i$

$$* = \sum_i^M p_{i,j} s_i$$

Дакле, вероватноћа да систем у  $n + 1$ -ом тренутку буде у стању  $j$  једнака је суми производа вероватноћа да се систем у  $n$ -том тренутку нађе у било ком стању и вероватноћа одговарајућих прелаза.

Ова сума представља  $j$ -ту колону у матрици ( димензија  $1 \times M$  ) која се добије при множењу вектора  $\vec{s}$  и матрице транзиције  $P$ .

Према свему наведеном следи да је закон расподеле случајне променљиве  $X_{n+1}$  ( расподела вероватноћа да се систем у  $n + 1$ -ом тренутку налази у сваком од стања ) једнак  $\vec{s} \times P$ .

Аналогно, у тренутку  $n+2$ , случајна променљива  $X_{n+2}$  има расподелу  $\vec{s} \times P^2$ , у тренутку  $n + 3$ , случајна променљива  $X_{n+3}$  има расподелу  $\vec{s} \times P^3$  итд.

**Дефиниција 10** *Расподела  $\vec{s}$  за коју важи :*

$$\vec{s} \times P = \vec{s}$$

*назива се **стационарна** или **равнотежна** расподела.*

$\vec{s} \times P$  представља "један корак у будућност", тј. расподелу вероватноћа да систем нађе у сваком од стања у следећем временском тренутку. Уколико расподела остаје иста, односно, вероватноће се не мењају са временом, тада се та расподела назива стационарном. Под одређеним условима везаним Марковљеве ланце, доказује се да Марковљев ланац **увек** конвергира ка својој стационарној расподели без обзира на полазно стање. Више о конвергенцији Марковљевих ланаца може се наћи у [11]. Дакле, полазне стање се може изабрати потпуно случајно а затим, уколико се дозволи да "протекне" довољно времена, закон расподеле вероватноће да се систем нађе у свим стањима система ће конвергирати ка стационарној расподели тог ланца.

**Дефиниција 11** *МСМС ( енгл. Markov Chain Monte Carlo ) методе представљају класу алгоритама који се користе за синтетичко генерисање узорака случајних променљивих из одговарајућих расподела. Овим методама се креирају Мерковљеви ланци који као равнотежну расподелу имају расподелу из које се узимају узорци. Једна од МСМС метода је и Гибсово узорковање ( енгл. Gibbs sampling )*

### Гибсово узорковање

Нека је дата заједничка расподела (енгл. joint distribution)  $p(\mathbf{z}) = p(z_1, z_2, \dots, z_M)$  из које је потребно одабрати неку вредност (енгл. sample ) и нека је познато почетно стање Марковљевог ланца који је потребно генерисати. Сваки корак Гибсовог узорковања почиње заменом вредности једне променљиве  $z_1, z_2, \dots, z_M$  вредношћу која се добија из **условне расподеле** те променљиве у односу на све остале. Дакле,  $z_i$  се мења вредношћу која се узима из расподеле  $p(z_i \mid z_{-i})$ , где је са  $z_i$  означена  $i$ -та координата вектора  $\mathbf{z}$  а са  $z_{-i}$  сви  $z_1, z_2, \dots, z_M$  без  $z_i$ . Ова процедура се наставља за све променљиве по неком одређеном редоследу. При довољном броју итерација, вредности вектора  $\mathbf{z}$  ће конвергирати ка  $p(\mathbf{z})$ .

На пример, нека је дата расподела три случајне променљиве  $p(z_1, z_2, z_3)$  и нека су вредности у тренутку  $t : z_1^t, z_2^t, z_3^t$ . Нека се замена вредности променљивих врши у односу на индекс, од најмањег ка највећем. Вредност  $z_1^t$  се мења новом вредношћу  $z_1^{t+1}$  која се узима ( узрокује ) из расподеле

$$p(z_1 \mid z_2^t, z_3^t).$$

Сада се вредност  $z_2^t$  мења са вредношћу  $z_2^{t+1}$  која се узима из расподеле

$$p(z_2|z_1^{t+1}, z_3^t).$$

Дакле, одмах се користи нова вредност променљиве  $z_1$ . Коначно, за промену вредности  $z_2^t$  користи се вредност  $z_3^{t+1}$  која се добија из расподеле :

$$p(z_2|z_1^{t+1}, z_3^{t+1}).$$

Овим је завршена **једна итерација** Гинбсовог узорковања. Исти процес се наставља кроз низ итерација све до одређеног броја или до неког другог услова заустављања.

Описана процедура се може уопштити и на више од три променљиве и може се представити следећим псеудокодом:

```

1. Initialize  $\{z_i : i = 1, \dots, M\}$ 
2. For  $\tau = 1, \dots, T$ :
  - Sample  $z_1^{(\tau+1)} \sim p(z_1|z_2^{(\tau)}, z_3^{(\tau)}, \dots, z_M^{(\tau)})$ .
  - Sample  $z_2^{(\tau+1)} \sim p(z_2|z_1^{(\tau+1)}, z_3^{(\tau)}, \dots, z_M^{(\tau)})$ .
  -  $\vdots$ 
  - Sample  $z_j^{(\tau+1)} \sim p(z_j|z_1^{(\tau+1)}, \dots, z_{j-1}^{(\tau+1)}, z_{j+1}^{(\tau)}, \dots, z_M^{(\tau)})$ .
  -  $\vdots$ 
  - Sample  $z_M^{(\tau+1)} \sim p(z_M|z_1^{(\tau+1)}, z_2^{(\tau+1)}, \dots, z_{M-1}^{(\tau+1)})$ .

```

Слика 3.19: Псеудокод Гибсовог узорковања, преузето са [12]

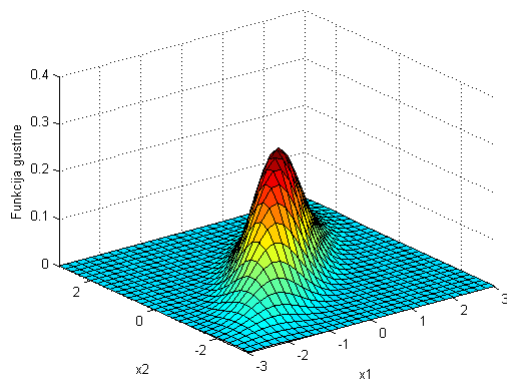
Гибсово узорковање подразумева да су унапред познате **условне расподеле** свих променљивих и да је могуће узорковање из њих.

*Пример :* Нека је потребно узорковати вредности из дводимензионалне нормалне расподеле  $\mathcal{N}(\mu, \Sigma)$  Гибсовим узорковањем при чему је

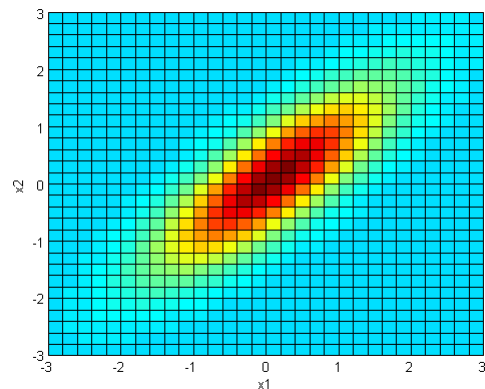
$$\mu = [\mu_1, \mu_2] = [0, 0]$$

$$\Sigma = \begin{bmatrix} 1 & \rho_{12} \\ \rho_{21} & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

Графички приказ овакве дводимензионалне нормалне расподеле дат је на следећој слици ( тродимензионално и пројектовано на две димензије ):



Слика 3.20: Тродимензионални приказ дводимензионалне нормалне расподеле



Слика 3.21: Дводимензионални приказ дводимензионалне нормалне расподеле

Основна претпоставка Гибсовог узорковања је да су познате условне расподеле свих променљивих и да је из њих могуће узорковати. Према [12] и [13], за условне расподеле дводимензионалне заједничке расподел важи :

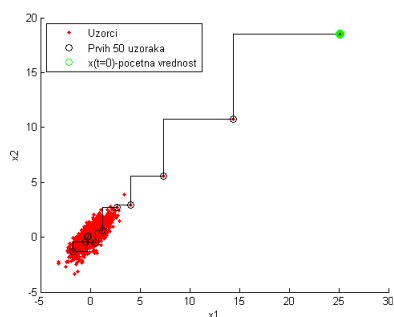
$$p(x_1 | x_2^{(t-1)}) = \mathcal{N}(\mu_1 + \rho_{21}(x_2^{(t-1)} - \mu_2), \sqrt{1 - \rho_{21}^2})$$

и

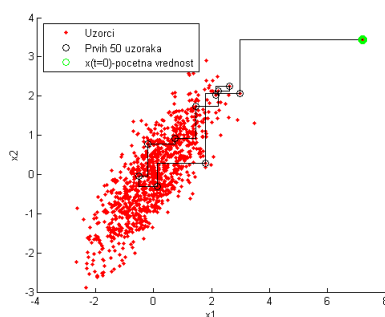
$$p(x_2 | x_1^{(t)}) = \mathcal{N}(\mu_2 + \rho_{12}(x_1^{(t)} - \mu_1), \sqrt{1 - \rho_{12}^2})$$

Дакле, обе условне расподеле представљају једнодимензионалну нормалну расподелу са одговарајућим параметрима. Почетне вредности променљивих се бирају случајно зато што нису од важности. Марковљев ланац ће свакако конвергирати ка дводимензионалној нормалној расподели са неведеним параметрима после одређеног броја итерација. У зависности од полазног стања, тај број итерација ће бити мањи или већи.

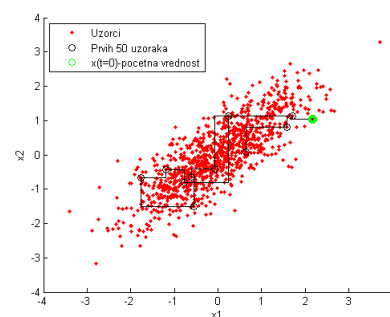
На следећем сликама су представљене добијене расподеле Гибсовим узорковањем за различите почетне вредности :



Слика 3.22: Почетна тачка (25.1284, 18.5165)



Слика 3.23: Почетна тачка (7.2162, 3.4380)



Слика 3.24: Почетна тачка (2.1649, 1.0314)

Са слика је очигледно да се првих неколико узорака може занемарити (3.22 може се занемарити првих 7-8 узорака).

Важно је приметити да се узимање узорака увек креће по "степенастом" обрасцу. Дакле, две суседне тачке имају исту једну координату ( $x$  или  $y$ ). То је зато што Гибсово узорковање у једном тренутку мења **само једну** променљиву у односу на одговарајућу вредност друге.

Конвергенција алгоритма Гибсовог узорковања ка стационарној расподели Марковљевог ланца је теоретски загарантована, али је у пракси јако тешко одредити број итерација након којих ланац почиње да конвергира. Један од начина процене конвергенције је и рачунање *log-likelihood* -а

### 3.4 Како ради ТМ алгоритам

Раније је неформално описан LDA генеративни процес. Основна претпоставка је да се сваки документ у одређеној пропорцији говори о свакој теми ( има одређену расподелу над темама) као и да свакој теми све речи из корпуса припадају са различитим вероватноћама ( расподела над речима). Генеративни процес се , према [14], може описати следећим псеудокодом :

1. For  $k = 1 \dots K$ :
  - (a)  $\phi^{(k)} \sim \text{Dirichlet}(\beta)$
2. For each document  $d \in \mathbf{D}$ :
  - (a)  $\theta_d \sim \text{Dirichlet}(\alpha)$
  - (b) For each word  $w_i \in d$ :
    - i.  $z_i \sim \text{Discrete}(\theta_d)$
    - ii.  $w_i \sim \text{Discrete}(\phi^{(z_i)})$

Слика 3.25: Генеративни процес LDA-a

при чему је :  $K$  - укупан број тема у колекцији  $\phi^{(k)}$  - расподела над свим речима из колекције и представља расподелу над речима у  $k$ -тој теми  $\theta_d$  - расподела над темама у документу  $d$ .  $z_i$  - тема којој припада реч  $w_i$ .  $\alpha, \beta$  - **хиперпараметри** тј. параметри симетричних Дирехлеових расподела.

Описани генеративни процес резултује формирањем следеће заједничке расподеле :

$$p(w, z, \theta, \phi \mid \alpha, \beta) = p(\phi \mid \beta) p(\theta \mid \alpha) p(z \mid \theta) p(w \mid \phi_z) \quad (3.16)$$

Непозанте променљиве које је потребно "открити" су  $z, \theta$  и  $\phi$  на основу (једино) познатих **речи** и њиховог присуства у сваком од докумената. Дакле, потребно је пронаћи расподеле наведених променљивих **под условом** да су познате речи и њихова распоређеност по документима тј. открити њихове постериорне расподеле. Основни проблем ТМ је **постериорно закључивање** (енг. posterior inference) односно отривање постериорних расподела латентних случајних променљивих на основу задатог скупа докумената и речи што представља решавање следеће једначине :

$$p(\theta, \phi, z \mid w, \alpha, \beta) = \frac{p(\theta, \phi, z, w, \mid \alpha, \beta)}{p(w \mid \alpha, \beta)} \quad (3.17)$$

Према [14], рачунање имениоца овог разломка је готово немогуће па се стога прибегава апроксимативним методама каква је и Гибсово узорковање.

Да би се применило Гибсово узорковање, потребно је познавати условне расподеле свих променљивих из чије се заједничке расподеле узоркује. Међутим, показује се да је довољно пронаћи само  $z$  јер се остале две променљиве могу преко ње израчунати и то (према [14]):

$$\theta_{d,z} = \frac{n(d, z) + \alpha}{\sum_{|Z|} n(d, z) + \alpha}$$

$$\phi_{z,w} = \frac{n(z, w) + \beta}{\sum_{|W|} n(z, w) + \beta}$$

Овако примењен алгоритам Гибсовог узорковања назива се још и енгл. Collapsed Gibbs Sampling. Дакле, циљ је пронаћи за сваку реч, вероватноћу да припадне свакој од тема, под условом да су познате теме којима припадају остале речи у том тренутку. Формалније, ово се може записати  $p(z_i \mid z_{-i}, \alpha, \beta, w)$  где  $z_{-i}$  представља доделу тема свим речима сем  $i$ -те.

Априорне расподеле коришћене у ТМ су Дирихлеове. Важна особина Дирихлеове расподеле је да је она **конјугована** са мултиномијалном расподелом. Дирихлеова расподела је расподела над параметрима мултиномијалне расподеле. Нека је на почетку претпостављено да параметри мултиномијалне расподеле припадају некој Дирихлеовој расподели -  $\mathbf{p} \sim \text{Dir}(\mathbf{p}, \alpha)$ . Нека је  $\mathbf{x}$  узорак генерисан из мултиномијалне расподеле  $\text{Mult}(\mathbf{x}; \mathbf{p})$ . Тада важи да је постериорна



расподела  $\mathbf{p}$  ( дакле, расподела под условом да је познат узорак  $\mathbf{x}$  ) такође **Дирихлеова расподела** са параметром  $\mathbf{x} + \alpha$  тј.:

$$p(\mathbf{p} \mid \mathbf{x}, \alpha) = \text{Dir}(\mathbf{p}; \mathbf{x} + \alpha) = \frac{1}{B(\mathbf{x} + \alpha)} \prod_{i=1}^{|\alpha|} p_i^{x_i + \alpha_i - 1} \quad (3.18)$$

Како (3.18) представља **расподелу** то важи да је :

$$1 = \int \frac{1}{B(\mathbf{x} + \alpha)} \prod_{i=1}^{|\alpha|} p_i^{x_i + \alpha_i - 1} = \frac{1}{B(\mathbf{x} + \alpha)} \int \prod_{i=1}^{|\alpha|} p_i^{x_i + \alpha_i - 1} \quad (3.19)$$

Одакле следи да је :

$$\int \prod_{i=1}^{|\alpha|} p_i^{x_i + \alpha_i - 1} = B(\mathbf{x} + \alpha) \quad (3.20)$$

Ова једнакост је важна за даљи опис рада ТМ алгоритма.

Према формули условне расподеле, важи :

$$p(z_i \mid z_{-i}, \alpha, \beta, w) = \frac{p(z_i, z_{-i}, w \mid \alpha, \beta)}{z_{-i}, w \mid \alpha, \beta)} \propto p(z_i, z_{-i}, w \mid \alpha, \beta) = p(z, w \mid \alpha, \beta) \quad (3.21)$$

$p(z, w \mid \alpha, \beta)$  се може посматрати као "маргиналан расподела" две променљиве заједничке расподеле (3.16) па важи :

$$p(z, w \mid \alpha, \beta) = \iint p(z, w, \theta, \phi \mid \alpha, \beta) d\theta d\phi = \iint p(\phi \mid \beta) p(\theta \mid \alpha) p(z \mid \theta) p(w \mid \phi_z) d\theta d\phi \quad (3.22)$$

Груписањем по заједночкој зависној променљивој, претхонда једначина се може написати :

$$p(z, w \mid \alpha, \beta) = \int p(z \mid \theta) p(\theta \mid \alpha) d\theta \int p(w \mid \phi_z) p(\phi \mid \beta) d\phi \quad (3.23)$$

Оба интерграла представљају комбинацију узорка из мултиномијалне расподеле и априорне Дирихлеове расподеле. Како је Дирихлеова расподела конјугована (conjugate prior) са мултиномијалном, у подинтегралном изразу се налази "множење" две Дирихлеове расподеле са одговарајућим параметрима.

Дакле : Пошто  $p(z \mid \theta)$  има мултиномијалну дистрибуцију, важи:

$$p(z \mid \theta) = \prod_{i=1}^D \prod_{k=1}^K \theta_{d,k}^{\Omega_{d,k}} \quad (3.24)$$

, где је  $\Omega_{d,k}$  број који означава колико пута је тема  $k$  додељена у документу  $d$  - број речи који у документу  $d$  припадају теми  $k$ .

Члан  $p(\theta \mid \alpha)$  је из основне Дирихлеове расподеле па важи :

$$p(\theta \mid \alpha) \stackrel{(1)}{=} \prod_{i=1}^D p(\overline{\mathbf{q}}_d \mid \alpha) \stackrel{(2)}{=} \prod_{d=1}^D \frac{1}{B(\alpha)} \prod_{k=1}^K q_{d,k}^{\alpha_k - 1} \quad (3.25)$$

где је  $\overline{\mathbf{q}}_d$  расподела вероватноћа тема у документу  $d$ . Расподеле вероватноћа тема по документима су независне, па је зато могуће написати (1). Расподела тема по документу се узима из Дирихлеове расподеле па је зато могуће написати (2).

Према томе, први интеграл једнакости (3.23) се записује као :

$$\int p(z | \theta) p(\theta | \alpha) d\theta = \int \prod_{i=1}^D \prod_{k=1}^K \theta_{d,k}^{\Omega_{d,k}} \prod_{d=1}^D \frac{1}{B(\alpha)} \prod_{k=1}^K q_{d,k}^{\alpha_k-1} d\theta_d \stackrel{(1)}{=} \prod_{i=1}^D \int \frac{1}{B(\alpha)} \prod_{k=1}^K q_{d,k}^{\Omega_{d,k}+\alpha_k-1} d\theta_d \quad (3.26)$$

Једнакост (1) следи из чињенице да су  $\theta_d$  независне расподеле па се могу интегралити посебно - правило интеграције производа Према релацији (3.20) претходна једнакост се може написати и као се :

$$\int p(z | \theta) p(\theta | \alpha) d\theta = \prod_{i=1}^D \frac{B(\Omega_d + \alpha)}{B(\alpha)} \quad (3.27)$$

где је са  $\Omega$  означена матрица докумената и тема,  $\Omega_{d,k}$  означава колико је пута тема  $k$  додељена у документу  $d$  а  $\Omega_d$  је  $d$ -та врста те матрице. Елементи ове матрице могу се математички записати и овако :

$$\Omega_{d,k} = \sum_{i=1}^N I(d_i = d \wedge z_i = k) \quad (3.28)$$

где је  $N$  укупан број речи у корпусу (са понављањем).

Аналогно претходним извођењима, и други интеграл може да се упрости:

Члан  $p(\phi | \beta)$  је из основе Дирхлеове расподеле па важи :

$$p(\phi | \beta) = \prod_{k=1}^K p(\phi_k | \beta) = \prod_{k=1}^K \frac{1}{B(\beta)} \prod_{v=1}^V \phi_{k,v}^{\beta_v-1} \quad (3.29)$$

Члан  $p(w | \phi_z)$  има мултиномијалну расподелу па важи :

$$p(w | \phi_z) = \prod_{i=1}^N p(\phi_{z_i, w_i}) = \prod_{k=1}^K \prod_{v=1}^V \phi_{k,v}^{\Psi_{k,v}} \quad (3.30)$$

где је  $\psi$   $K \times V$  матрица а  $\psi_{k,v}$  броји колико тема  $k$  била додељена речи  $v$ . Ова матрице може се још написати као :

$$\psi_{k,v} = \sum_{i=1}^N I(w_i = v \wedge z_i = k) \quad (3.31)$$

На основу (3.29) и (3.30) следи :

$$\int p(w | \phi_z) p(\phi | \beta) d\phi = \int \prod_{k=1}^K \frac{1}{B(\beta)} \prod_{v=1}^V \phi_{k,v}^{\psi_{k,v}+\beta_v-1} d\phi_k \quad (3.32)$$

Аналогно извођењу (3.26) (3.27) следи :

$$\int \prod_{k=1}^K \frac{1}{B(\beta)} \prod_{v=1}^V \phi_{k,v}^{\psi_{k,v}+\beta_v-1} d\phi_k = \prod_{k=1}^K \left( \frac{1}{B(\beta)} \int \prod_{v=1}^V \phi_{k,v}^{\psi_{k,v}+\beta_v-1} d\phi_k \right) = \prod_{k=1}^K \frac{B(\psi_k + \beta)}{B(\beta)} \quad (3.33)$$

Коришћењем једнакости (3.27) и (3.33), једнакост (3.23) се може записати као :

$$p(z, w | \alpha, \beta) = \prod_{i=1}^D \frac{B(\Omega_d + \alpha)}{B(\alpha)} \prod_{k=1}^K \frac{B(\psi_k + \beta)}{B(\beta)} \quad (3.34)$$

На основу (3.34) може се извести правило по коме ће Гибсов алгоритам узорковања мењати доделе тема речима. Дакле :

$$p(z_i = k | Z^{-i}, W, \alpha, \beta) = \frac{p(z_i = k, Z^{-i}, W | \alpha, \beta)}{p(Z^{-i}, W | \alpha, \beta)} = \frac{p(Z, W | \alpha, \beta)}{p(Z^{-i}, W | \alpha, \beta)} \quad (3.35)$$

Именилац претходне једнакост се може написати преко условне вероватноће у следећем облику :

$$p(Z^{-i}, W | \alpha, \beta) = p(Z^{-i} | \alpha\beta)p(W | Z^{-i}, \alpha\beta) \stackrel{(1)}{=} \quad (3.36)$$

$$= p(Z^{-i})p(W^{-i} | Z^{-i})p(w_i) \propto p(Z^{-i})p(W^{-i} | Z^{-i}) = p(Z^{-i}, W^{-i}) \quad (3.37)$$

Једнакост (1) следи из чињенице да свако  $z_i$  зависи само од  $w_i$ . Од ове једнакости параметри  $\alpha, \beta$  су изостављени због прегледсности, али се подразумевају.

Форма једнакости (3.37) иста је као (3.31) па се једнакост (3.35) записује као :

$$p(z_i = k | Z^{-i}, W, \alpha, \beta) = \prod_{k=1}^K \frac{B(\psi_k + \beta)}{B(\psi_k^{-i} + \beta)} \prod_{d=1}^D \frac{B(\Omega_d + \alpha)}{B(\Omega_d^{-i} + \alpha)} \quad (3.38)$$

Коришћењем особина бета фунцкије, претходна једнакост се своди на :

$$p(z_i = k | Z^{-i}, w = v, W^{-i}, \alpha, \beta) = \frac{\psi_{k,v} + \beta_{w_i} - 1}{\left[ \sum_{v=1}^V \psi_{k,v} + \beta_v \right] - 1} [\Omega_{d,k} + \alpha_k - 1] \quad (3.39)$$

Детаљно извођење може се наћи код [14] и [15].

### 3.4.1 Имплементација - псеудокод

**Input:** words  $\mathbf{w} \in$  documents  $\mathbf{d}$   
**Output:** topic assignments  $\mathbf{z}$  and counts  $n_{d,k}, n_{k,w}$ , and  $n_k$

```

begin
  randomly initialize  $\mathbf{z}$  and increment counters
  foreach iteration do
    for  $i = 0 \rightarrow N - 1$  do
      word  $\leftarrow w[i]$ 
      topic  $\leftarrow z[i]$ 
       $n_{d,topic} = 1$ ;  $n_{word,topic} = 1$ ;  $n_{topic} = 1$ 
      for  $k = 0 \rightarrow K - 1$  do
         $p(z = k | \cdot) = (n_{d,k} + \alpha_k) \frac{n_{k,w} + \beta_w}{n_k + \beta \times W}$ 
      end
      topic  $\leftarrow$  sample from  $p(z | \cdot)$ 
       $z[i] \leftarrow$  topic
       $n_{d,topic} = 1$ ;  $n_{word,topic} = 1$ ;  $n_{topic} = 1$ 
    end
  end
  return  $\mathbf{z}, n_{d,k}, n_{k,w}, n_k$ 
end

```

**Algorithm 1:** LDA Gibbs Sampling

Слика 3.26: Псеудокод - преузето са [15]

## Глава 4

# Припрема подарака

Подаци кориштени за тестирање приликом израде овог рада су јавно доступни подаци са сајта <http://stackexchange.com/> из три области - инжињерство, фитнес и хемија. Подаци су на **енглеском језику**.

Ове три области су намерно тако одабране како би се повећала разноврсност речи. Речи из сродних научних грана користе сличну или исту терминологију па опасност од бирања питања и одговора из сродних научних дисциплина лежи у чињеници да ће диверзитет корпуса бити мали.

Из сваке области узето је по 200 питања и одговора, што представља базу од укупно 600 питања и 600 одговора. Овакав скуп података је затим подељен на два дела - тренинг део (360 питања и одговора) и тест део (240 питања и одговора).

Након испитивања на овом скупу података, решење је тестирано и на подацима доступним са сајта [answers.yahoo.com](http://answers.yahoo.com). На овај начин, „дуплим“ тестирањем, обезбеђује се независност закључака од природе података.

Подаци добијени са означених сајтова нису погодни за директну обраду те их је потребно делимично **прерадити** тј. **предпроцесирати**.

Надаље следи опис коришћених метода предпроцесирања. Приликом тестирања коришћене су различите комбинације ових метода и мерени су њихови утицаји на крајње решење.

### 4.1 Уклањање HTML ознака и неалфанумеричких карактера

Подаци који су доступни преко сајта <http://stackexchange.com/> дати су у форми HTML текстова. То значи да се кроз основни текст питања и одговора провлаче додатни низови карактера који представљају HTML ознаке (тагове). Ове ознаке добијају смисао приликом генерисања веб странице, односно приликом форматирања текста питања и одговора на веб страницама. Међутим, за обраду текста у овом раду, те ознаке не значе ништа и додатно могу уносити забуну. Због тога их је потребно, пре било какве даље обраде текста, склонити. Најједноставнији начин да се то уради је преко *регуларних израза*, обзиром на изглед и формат HTML ознака.

Са друге стране, у тексту се поред слова и бројева (*алфанумерички карактери*) обавезно појављују и специјални знаци (ознаке које нису ни слова ни бројеви - *неалфанумерички карактери*, међу којима су најчешћи интерпункцијски знакови). Поред њих, у скорије време, уз сваки текст је готово обавана појава и група специјалних знакова као што су нпр. смајлићи (структуре конструисане од специјалних знакова које се могу интерпретирати као расположење нпр. :) - срећа, :( - туга итд.). Са становишта алгорита моделовања тема, сви они, исто као и HTML ознаке, немају значење и потребно их је уклонити. Такође, најједноставнији начин да се то уради је применом регуларних израза.

## 4.2 Конвертовање свих слова текста у „мала слова“ енг. lowercase

Писање речи великим или малим словима, као и започињање речи малим или великим словом, углавном има граматички смисао. При томе, у већини случајева, реч написана почетним великим и иста реч написана почетним малим словом, имају исто значење. У машинској обради података, свако слово има другачију бинарну репрезентацију. Стога, реч написана великим почетним словом и иста реч написана малим почетним словом или свим великим словима, **нису исте речи** обзиром на то да имају различиту репрезентацију. Како би се ово избегло, пре било какве даље обраде, врши се конвертовање или пребацивање свих слова текста у мала слова. На тај начин, све речи састављене од истих слова и у истом редоследу, представљају **исту** реч, без обзира на граматичко значење речи или њену позицију у тексту. Избор конвертовања у мала слова је једнако оправдан као и конвертовање у велика слова. Дакле, исти резултат би се добио и конвертовањем свих слова у велика слова. Међутим, у пракси, је чешће прихваћено пребацивање у мала слова па је такав приступ усвојен и у овом раду. Наравно, постоје бројни примери када писање речи великим или малим почетним словом битно мења значење речи. На пример, у српском језику, реч *Мила*, написана великим почетним словом, означава име особе, именуцу, док реч *мила*, написана малим почетним словом означава придев. Исто тако реч, *Јела* односи се на име особе док се реч *јела* односи на врсту зимзеленог дрвета. Међутим, оваквих речи има довољно мало да је ризик од мењања значења речи, са аспекта алгорита моделовања тема, прихватљив.

## 4.3 Издајање атомских елемената докумената - токена, енг. tokenization

Манипулација читавим документима са аспекта алгорита моделовања тема, нема смисла. Основна јединица манипулације ове врста алгоритама је **реч**. Према томе, потребно је документе рашчланити на појединачне речи. Процес издајања основних елемената манипулације, тј. елемената од интереса, назива се издајање **токена** или **токенизација**. Обзиром да су овде атомски елементи **речи** и да су речи одвојене размаком, процес токенизације је најједноставније извршити преко регуларних израза. У пакету Mallet се већ налазе готове класе

## 4.4 Избацавање често коришћених речи енг. stop words

У свакодневном говору често се употребљавају личне заменице, прилози, везници итд. Без њих, говор би био неодређен и неповезан па самим и неразумљив. Међутим, у машинској обради текстуланих података, поготово у алгоритмима моделовања тема, оваква врста информација није неопходна. Пре свега, такве речи не носи суштинско теметско значење обзиром да нису уско повезана ни са једном конкретном области. На пример, везници ,као што су у српском језку: и, или, па, али, због, ради итд. се употребљавају при писању докумената из свих научних грана и стога се ни за једну од тих речи не може дефинисати област припадања. Обзиром да је циљ из групе докумената издвигнути **теме**, овакве речи су сувишне. Штавише, уносе додатну забуну при закључивању и, обзиром да су бројене, могу представљати велико оптерећење приликом обраде. Подаци коришћени у при изради овог рада су на енглеском језику те ће се надаље говорити о оваквој врсти речи у енглеском језику. Међутим, узимањем уобзор специфичности конкретног језика, иста разматрања могу се применити и на друге језике.

Изабацивање често коришћених речи, stop words-ова, може се реализовати на више начина.

Поузданији, универзалнији али и рачински захтевнији начин је алгоритамско проналажење таквих речи. Обзором да немају тематско значење, оне се појављују у великом броју у свим документима и темама. Једноставним бројањем појављивања речи у скупу свих докумената могу се уочити групе речи које се са изузетно високим фреквенцијама јављају у **свим документима**. Такве речи се могу сматрати за често коришћене те се, из описаних разлога, избацују. Поред једноставног бројања речи, постоје и друге методе „ мерења “ присуства речи у корпусу. Једна од њих је и релативна фреквенција која зависи од дужине докумената. Међутим, обзором да овај приступ није коришћен у раду, у ове методе се неће дубље улазити.

Други, мање поуздан и релативно рачуноски незахтеван приступ је коришћење **листе често коришћених речи** које постоје за сваки језик. Те листе су јавно доступне и могу се пронаћи на бројним веб сајтовима. Обзором да је циљ рада био истраживање примене алгоритама моделовања тема у специфичном проблему, овај приступ је прихватљивији. Пре свега, релативно се лако имплементира обзором да у Mallet-у већ постоји класа за уклањање ових речи. Са друге стране, овај корак предпроцесирања се на овај начин претвара у тривијалан и оставља простор за истраживање самог алгорита моделовања тема. Највећа опасности од овог приступа је елиминација речи које, иако сврстане међу често коришћене, у датом скупу докумената ипак имају значење. Исто тако, обзиром да је листа предефинисана, могуће је изоставити речи које у конкретном скупу представљају често коришћене речи.

За поузданије и детаљнија истраживања, предлаже се примена прве методе.

## 4.5 Додавање синонима

У циљу бољег дифренцирања тематике питања и одговора, за сваку реч је додато по 5 синонима. За проналажење синонима је коришћена WordNet библиотека. Основни разлог додавања синонима у скуп била је претпоставка да ће се на тај начин боље диференцирати теме, повећати диверзитет корпуса а самим тим и олакшати препознавање тачног одговора. Међутим, резултати су показали управо супротно. Разлог томе што синониме треба тражити **по смислу** речи а не само по лексичком облику речи. Такође, фразе, којих има доста у свакодневном говорном и писаном енглеском језику, значајно могу да утичу на смисао питања/одговора. Када се они рашчлане на појединачне речи, могуће је да се и смисао промени.

## 4.6 Склањање наставака речи - енг. *stemming*

Овај сегмент предпроцесирања је изузетно завистан од језика на коме је текст писан. Циљ је препознати различите облике исте речи и свести је на заједничку основу, која **не мора** да буде коренска. У енглеском језику, различити облици речи граде се додавањем разних **наставака** као што су *s,ing,es ....* Дакле, склањањем наставака речи редукује се диверзитет корпуса али се истовремено речи које имају исто значење само различит облик настао услед контекста реченице - рода, времена, врсте речи итд, свде на исту реч. У конкретној примени, овако нешто је неопходно за прецизно раздвајање тема. За српски језик овако нешто не би било могуће имплементирати на једноставан начин обзором на промену речи по падежима, лицима ( за глаголе ), родовима, бројевима и гласовне промене које се при томе дешавају. У тренутку писања рада, никакво готово решења за српски језик није постојало. Уколико предмет истраживања неког будућег рада буде био српски језик, потребно је написати процедуре којима се речи језика ослобађају наставака, следећи граматичка правила.

## 4.7 Свођење на коренску реч - енг. lemmitization

Свођење речи на коренску реч је слична метода методи склањања наставака речи, такође уско повезана са језиком који се обрађује. Једна од разлика је што се склањање наставака речи може применити на речи које мењају облик додавањем наставака док се свођење на коренску реч може применити на све речи. На пример, реч енг. better - бољи, при склањању наставака би остала непромењена или би се свела на реч енг. bett, док при свођењу на коренску реч она постаје енг. good - добар. Друга, битнија разлика, је што се уклањање наставака примењује на речи не водећи рачуна о контексту, док се свођењем на коренску реч може специфицирати и контекст речи. На пример, реч енг. meeting - може имати више значења. Као именица она означава *састанак* док као глагол означава презент партицип глагола *to meet*, у смислу *сретати се*, . Уклањањем наставака, реч енг. meeting у контексту именице као и у контексту глагола биће сведена на реч енг. meet. Код свођења речи на коренску реч, спецификацијом врсте речи могуће је реч енг. meeting оставити непромењену. Такође, свођењем на коренску реч могуће је неправилне глаголе енглеској језика свести на основни облик, што склањањем наставака није било могуће. Са аспекта алгоритма моделовања тема, свођење на коренску реч је прихватљивија метода, пре свега због могућности препознавања различитог облика истих речи када се они не граде додавањем наставака. Обзором на једноставност имплементације и брзину рада, чешће се користи скалањање наставка од свођења на коренску реч. У конкретном раду, обе методе су независно тестиране и равноправно коришћене у циљу добијања бољих резултата.

## Глава 5

# Решење проблема применом алгоритма моделовања тема

Проналажење правог одговора на постављено питање може бити изузетно сложен проблем чак и за човека. Оно што је суштински важно за препознавање ваљаног одговора је разумевање **суштине** односно смисла питања. За разлику од машина, човек на основу знања, уме да наслути тај смисао а самим тим и да препозна адекватан одговор. Међутим, уколико би се пред човека ставило питање из области о којој он нема никаквог знања ( не разуме значење речи) или је на језику који не разуме, врло је вероватно да би препознавање правог одговора било јако непоуздано. Са друге стране, немогуће је не запитати се шта заправо представља прави одговор на постављено питање. Данас је можда лакше него икада поставити питање и у кратком времену добити велики број одговора од различитих корисника. Учесници конверзације не морају нужно бити стручњаци из области које се тиче питања. Исто тако, велики број одговора, иако су наизглед адекватни, наилазе на осуду стручне популације. Дакле, потребно је извесно време, у коме долазо до комуникације међу различитим корисницима ( давање оцена се може сматрати комуникацијом) да би се **закључило** шта је прави одговор. Сајтови који су служили као извор података су управо описаног карактера. Дакле, одговори на постављено питање се оцењују од стране заинтересованих корисника и након неког времена са великом прецизношћу се може рећи који је адекватан одговор на постављено питање. Према томе, подаци који су овде разматрани јако су зависно од :

- атрактивности теме којом се баве - што је тема популарнија то ће већи број корисника бити укључен у давање и оцењивање одговора. Самим тим, може се сматрати да атрактивније теме имају поузданије одговоре
- природе питања - на нека питања се може одговорити врло кратко - (на пример где се ПМФ налази у Крагујевцу ) док друга питања захтевају опширне одговоре (нпр. детаљан опис историје Крагујевца или препричано књижевно дело)

У правом одговору не морају нужно да се нађу речи из питања. Исто тако, не мора постојати законитост између дужине питања и одговора. Према томе, не постоји **алгоритам** којим се може закључити који је адекватан одговор а да се при томе не укљичи додатно експертско знање.

Основна идеја овог рада је испитивање да ли се и у којој мери вештачно знање које се добија применом алгоритма моделовања тема може употребити за селектовање правог одговора без убацивања додатног експертског знања.



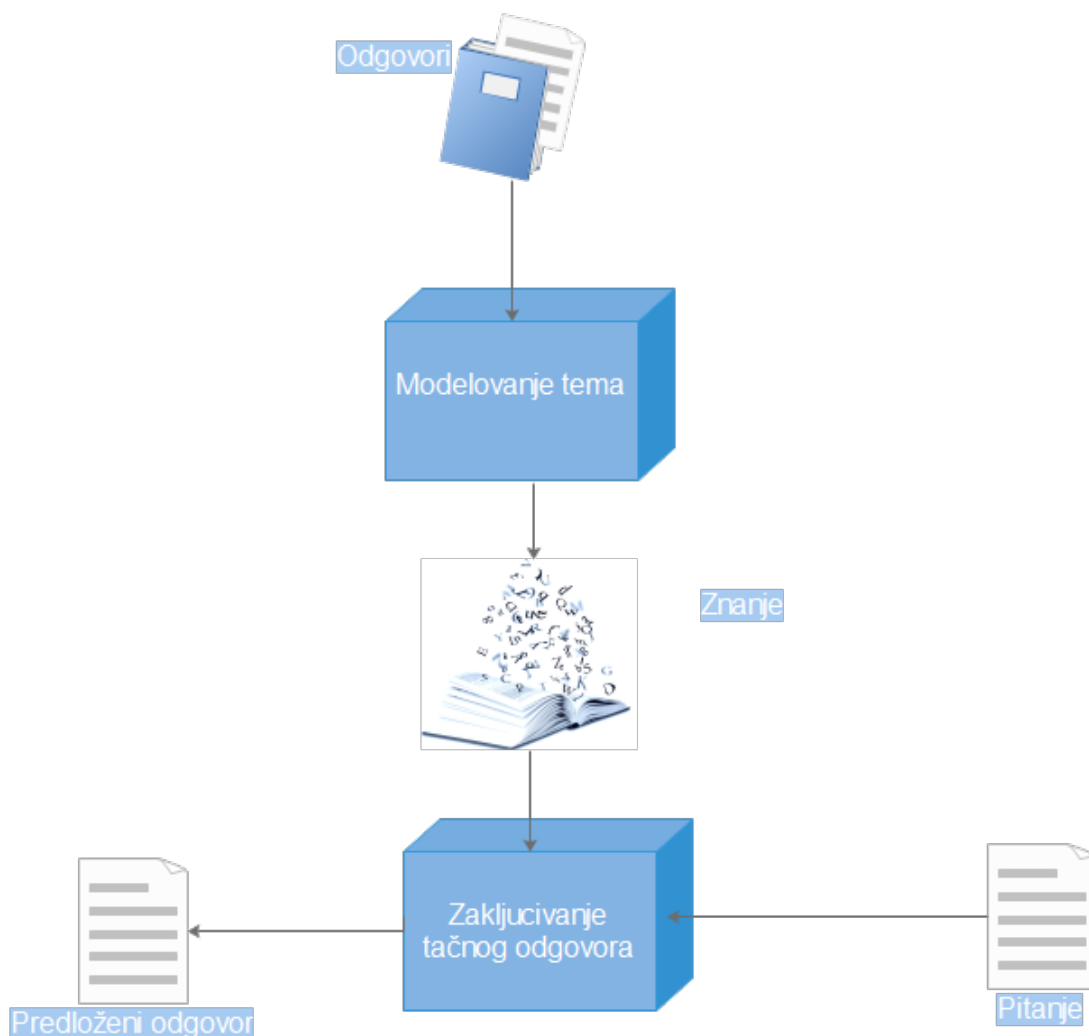
## 5.1 Опис решења

Идеја решења је изградња **модела тема** над свим одговорима чиме би се добио истрениран модел који поседује знање о том скупу докумената. Под знањем се подразумева расподела тема над документима као и расподела речи унутар тема. Основна претпоставка је да се при постављању питања то знање може употребити како би се селектовао тачан одговор. Даље се претпоставља да су питање и одговор највероватније из истих области ( једна или више ) односно да говоре о истим темама. Овде је важно напоменути да теримн **област** или **тема** више није упоредив са човековим схватањем тема или области. Обзорим да је број тема који се у истраживању користио јако велики - од 100 до 2000 - и да тема није ништа друго до скуп речи као и да у систем није укључено никакво додатно експертско знање, ово напомена је сасвим очигледна.

Селектовање правог одговора вршено је мерењем **сличности** између постављеног питања и свих одговора. Онај одговор који је **најсличнији** постављеном питању, одабира се као тачан одговор на постављено питање. Обзиром да је познато који одговор припада ком питању, рад програма је једноставно проверити и измерити.

За мерење сличности питања и одговора коришћено је неколико метода које се разликују по прецизности, брзини рада и меморијским захтевима.

Идејни ток решења може се представити дијаграмом као на слици 4.1:



Слика 5.1: Ток решења

## 5.2 Мерење сличности

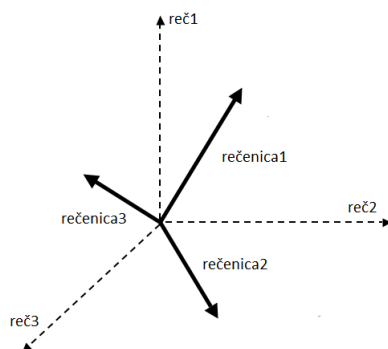
Мерење сличности је од суштинске важности за одабирање одговора на постављено питање. Она утиче на прецизност решења, брзину извршавања, меморијске захтеве итд. У раду су испитане три мере сличности од којих је мера заснована на вероватноћи дале најбоље резултате.

### 5.2.1 Косинусна сличност

Један од излаза алгоритама тема је и расподела тема по документима. Такође, за сваки нови документ, могуће је предвидети расподелу по темама. Дакле, пошто се вектор расподела по темама увек може направити, корисно их је искористити као меру сличности два документа.

Вектор расподеле по темама, дужине  $n$ , може се замислити као права у  $n$ -димензионаланом простору. Дакле, вектор питања и вектор одговора могу се замислити као две праве у  $n$ -димензионаланом. Што су те две праве „ближе“ једна другој, односно што је угао између њих ближи  $0$ , то су питање и одговор сличнији. Пошто се ради о расподелама, максимална вредност коју може да узме нека координата овако дефинисаног вектора је  $1$ , док је минимална вредност  $0$ , што значи да се обе праве налазе у првом квадранту. Дакле, максимални угао који два вектора могу да граде је  $90$  степени и, у смислу косинусне мере, означава да су документи потпуно различити. Близина праве питања и праве одговора означава сличну расподелу по темама. Ако се узме у обзир полазна претпоставка да питање и одговор „говоре о“ истим темама, постаје јасно због чега се угао између ових правих може узети за меру сличности два документа.

Ради илустрације, следи један прилично упрошћен и нереалан пример. Нека је скуп свих могућих речи састављен од три речи :  $reč1, reč2, reč3$  и нека су дате три реченице састављене од поменутих речи :  $rečenica1, rečenica2, rečenica3$ . Дате реченице могу се графички представити као праве ( Слика )



Слика 5.2: Графички приказ косинусне сличности

Угао између сваке две праве представља меру сличности реченица.

Уместо мерења угла између два вектора, практичније је мерити косинус тог угла. Косинус угла који заклапају два вектора може се одредити следећом формулом :

$$\cos\theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \|\vec{b}\|}$$

Претходна формула следи директно из дефиниције **скаларног производа** два вектора.

### 5.2.2 Мерење сличности према лексичкој и тематској сличности

Поред теметске сличности докумената, и лексичка сличност може бити важна. На пример, уколико се у одговору појављују исте речи као и у питању, **вероватније** је да је тај одговор ближи тачном одговору него одговор који нема заједничких речи са питањем. Наравно, могу се наћи примери у којима ово не важи. Али, исто тако, могуће је пронаћи примере у којима одговор и питање нису тематски слични - нпр. питање је уско специјализовано док је одговор теметски недефинисан. Према томе, знајући да обе мере сличности не морају увек да гарантују селектовање правог одговора, уз одговорајући ризик, може се испитати утицај комбинације ове две мере на одабир одговора.

Поставља се питање на који начин измерити лексичку сличност два документа тј. питања и одговора. Једно од решења би било једноставно бројање истих речи. Међутим, како је циљ испитати утицај **комбинације** лексичке и теметске сличности, ово решење се не може прихватити као добро. Разлог томе је што је лексичка сличност два документа измерена на овај начин **увек константна** док се теметска сличност докумената разликује зависно од параметара модела.

Други начин мерења лексичке сличности докумената може се добити мерењем лексичке сличности **тема**. Пошто се тема математички представља као функција расподеле над скупом свих речи, то сигурно за сваку реч из документа постоје вероватноће са којима свака тема садржи дату реч. Према томе, за сваки документ је могуће саставити  $K$  вектора, где је  $K$  укупан број тема, при чему  $i$ -ти вектор садржи вероватниће припадања речи документа  $i$ -тој теми.<sup>1</sup> Формалније  $i$ -ти вектор се може записати као :

$$p_q^{(i)} = p(w_q | i, \phi^{(i)}) \quad (5.1)$$

при чему је скуп речи документа дат са  $w_q = (w_1, w_2, \dots, w_{|q|})$  док је  $\phi^{(i)}$  расподела речи унутар  $i$ -те теме. Дакле, вектор  $p_q^{(i)}$  се формира тако што се за сваку реч из документа пронађе вероватноћа припадања те речи  $i$ -тој теми тј. важи да је  $p_q^{(i)} = (\phi_{w_1}^{(i)}, \phi_{w_2}^{(i)}, \dots, \phi_{w_{|q|}}^{(i)})$

Нека је одабрана тема  $i$ . Да би се измерила лексичка сличност питања и одговора унутар ове теме, потребно је формирати описане векторе за оба документа. Димензије ова два вектора треба да буду исте и једнаке укупном броју различитих речи у корпусу. При томе, вероватноће оних речи из корпуса које не припадају документу ће у векторима бити постављене на нулу. Исто тако, треба водити рачуна да редослед навођења речи ( тј. њихових вероватноћа) у оба вектора буде исти. То значи да ако је нпр. вероватноћа речи *математика* у првом вектору наведена на 3. позицији, тада се на 3. позицији у другом вектору налази такође вероватноћа за реч *математика*.

Сличност ова два вектора може да се мери на више начина. У раду је коришћена косинусна мера сличности док је у раду [17] коришћена Јенсен-Шенонова сличност.

Дакле, лексичка сличност два документа у  $i$ -тој теми може се представити као :

$$W(p_{pitanje}^{(i)}, p_{odgovor}^{(i)}) = kosinusna\_slicnost(p_{pitanje}^{(i)}, p_{odgovor}^{(i)})$$

Укупна лексичка сличност питања и одговора се може дефинисати као :

$$slicnost_1(pitanje, odgovor) = \frac{1}{K} \sum_{k=1}^K W(p_{pitanje}^{(k)}, p_{odgovor}^{(k)})$$

На овај начин је измерена лексичка мера сличности између питања и одговора. Даље, потребно је дефинисати тематску сличност питања и одговора. Она се дефинише као

<sup>1</sup>Ово се може гарантовати за сваку реч из одговора обзиром да је над скупом свих одговора изграђен модел. Међутим, може се догодити да се у питању појави реч која не постоји у корпусу модела. Вероватноће те речи у свим темама је тада 0

косинусна сличност расподеле тема унутар питања и одговора ( већ опшисана косинусна мера сличности). Дакле:

$$slicnost_2(pitanje, odgovor) = kosinusnaSlicnost(\theta_p pitanje, \theta_o odgovor))$$

Коначно, сличност питања и одговора се дефинише као :

$$slicnost(pitanje, odgovor) = slicnost_1(pitanje, odgovor) * slicnost_2(pitanje, odgovor)$$

Ова мера је у пракси показала јако добре резулате. Међутим, битан недостатак јој је јако велика сложеност. Најпре, за свако питање и за сваки одговор потребно је проћи кроз све теме. Ово је већ сложеност  $O(n^3)$  која је неприхватљива у оваквој врсти проблема. Поред временске сложености, и просторна сложеност овог решења није мала. Наиме, за сваку тему мора се чувати (или стално изводити) расподела по свим речима. Величина корпуса може бити јако велика, па је ове структуре готово немогуће чувати у меморији. Због велике сложености, ова мера није испитана детаљно као косинусна мера, па су резултати добијени овом мером прилично непоуздани и оквирни. Она није погодна за решавање оваквих типова проблема где величина уланог скупа може достићи и 20 000 докумената. Међутим, при мањем броју докумената, њена просторна и временска сложеност може да се компензује прецизношћу која се њома добија. Подробнија испитивања ове мере нису рађена у раду а више о једној варијанти ове мере може се наћи у [17].

### 5.2.3 Мерење сличности према предвиђеној вероватноћи

Главни резултати алгорита моделовања тема су две расподеле - расподела речи по темама и расподела тема по документима. Ове две расподеле могу да се употребе како би се одредила сличност два документа.

Нека је дат документ  $D$  и нека су познате  $\theta_D$ -расподела тема унутар тог документа и  $\phi$ -расподеле речи унутар свих тема. Тада се вероватноћа припадања неке речи  $w$  документу  $D$  може изразити као :

$$P_{lda}(w | D) = \sum_{z=1}^K P(w | z)P(z | \theta_D) \quad (5.2)$$

при чему :

$P(w | z)$  означава вероватноћу речи  $w$  унутар теме  $z$ . Обзиром да речи са различитим вероватноћама припадају различитим темама и да је хиперпараметар Дирихлеове расподеле за расподелу речи над темама  $\beta$ , ова вероватноћа се изражава као условна, под условом  $z$  и  $\beta$ . Хиперпараметар  $\beta$  се подразумева, обзиром на начин како је модел направљен, тако да се и при писању може изоставити.

$P(z | \theta_D)$ - означава вероватноћу са којом се тема  $z$  јавља у документу  $D$ . Из сличних разлога као и код претходног чиниоца, ова вероватноћа се изражава као условна под условом  $\theta_D$  и  $\alpha$ , стим што се  $\alpha$  подразумева па се и не пише.

Дакле, формулом (5.2) може се прерачунати колико је вероватно да реч  $w$  припада документу  $D$ . Ово се још може посматрати и као вероватноћа да је реч  $w$  генерисана документом  $D$ .

Имајући ово у виду, може се сада дефинисати и вероватноћа да скуп речи припада документу  $D$ , и то као :

$$P_{lda}(Q | D) = \prod_{w \in Q} P_{lda}(w | D) \quad (5.3)$$

Претходна једнакост устави дефинише вероватноћу да је скуп реч генерисан документом  $D$ . Узимајући специјално да је тај скуп речи питање које се поставља систему, једнакост се може протумачити и као вероватноћа генерисања постављеног питања из датог одговора.

**Вероватноћу генерисање** треба схватити као могућност извлачења речи питања из одговора. Што је ова вероватноћа већа, већа је и могућност да питање и одговор говоре о истим стварима те да посматрани одговор може бити тражени одговор на постављено питање.

Вероватноћа припадања речи  $w$  неком документу  $D$  се, поред формуле (5.2) може посматрати и из угла класичне вероватноће. Дакле, вероватноћа да ће се реч  $w$  наћи у документу  $D$  једнака је укупном броју појављивања речи  $w$  у том документу подељено са укупним бројем речи документа, односно :

$$P(w | D) = \frac{f_{w,D}}{|D|} \quad (5.4)$$

где је  $f_{w,D}$  број појављивања речи у документи док је  $|D|$  укупан број речи у документу.

Обзиром да реч  $w$  може бити било која реч, није нужно да она припада документу  $D$ . Дакле, може да се деси да ова вероватноћа буде 0. Имајући у виду формулу (5.3), овако нешто је апсолутно неприхватљиво, поготово код дужих питања. На пример, уколико постоји само једна реч која се налази у питању а не налази у одговору, док се осталих, рецимо, 100 поклапају, то би резултовало вероватноћом 0 за генерисање питања из текста одговора. Овако нешто, наравно, не може да буде тачно. Овај проблем може се решити увећањем **псеудо појављивања**. Псеудо појављивања представљају број појављивања који се узима као подразумевани уколико се реч питања не налази у тексту одговора. На овај начин свакој речи питања ће се доделити нека вероватноћа, која није 0, али је ипак довољно мала да ће велико непоклапање речи између два документа одразити на резултат. Псеудо појављивања могу се увести на више начина. У конкретном раду, формула (5.4) замењена је са формулом (5.5).

$$P(w | D) = \frac{f_{w,D} + \mu \frac{c_{w_i}}{|C|}}{|D| + \mu} \quad (5.5)$$

где је

$\mu$  параметар који се експериментално одређује и представља псеудо појављивање

$C$  је ознака укупног корпуса речи добијеног из свих одговора

$c_{w_i}$  - појављивање речи у корпусу.

Ни овакво решење није без мане. Наиме, и даље може да се деси да вероватноћа (5.3) буде једнака 0. То је случај када се у тексту питања појављује реч која није нашла **ни у једном** од одговора. Међутим, овакви случајеви су ретки, поготово код већих скупова улазних података. Међутим, ако се то и деси, таквој речи, уместо вероватноће рачунете на било који од описаних начина, се додељује вредност хиперпараметра  $\beta$ . Ова вредност узета је зато што се у алгоритму моделовања тема, управо она додељује свим речима у оквиру свих тема у нултој итерацији. Та вредност сигурно није нула, али је довољно мала како би утицала на резултат.

Једнакостима (5.5) и (5.2) дефинисане су вероватноће припадања неке речи одређеном документу али са различитим физичким смислом. Једнакост (5.2) дефинише тематску сличност док једнакост (5.5) дефинише лексичку сличност. Међутим, ове две сличности не морају увек да буду подједнако важне, иако су обе значајне. Због тога би требало да постоји могућност контролисања удела са којим ове две сличности улазе у крајњу процену сличности. У конкретном раду, овај проблем решен је додавањем додатног параметра  $\lambda$ . Што је његова вредност већа, то је лексичка сличност докумената важнија и обрнуто.

Помениуте вероватноће могу се уклопити тако да заједно граде меру сличности. У конкретном раду, вероватноћа генерисања речи из текста дата је са:

$$P(w \mid D) = \lambda \left( \frac{f_{w,D} + \mu \frac{c_{w_i}}{|C|}}{|D| + \mu} \right) + (1 - \lambda) \left( \sum_{z=1}^K P(w \mid z) P(z \mid \theta_D) \right) \quad (5.6)$$

Мера укупне сличности два документа дата се и даље рачуна преко (5.3). За вредности параметра  $\lambda$  и  $\mu$  могу се узети било које вредности, уз ограничење  $\lambda \leq 1$ . За потребе рада, експериментално су одређене вредности  $\mu = 200$  и  $\lambda = 0.2$ .

## Глава 6

# Развој решења

Циљ мастер рада био је *истраживање могућности* примене алгоритама моделовања тема у основној верзији при предлагању одговора на питање постављено природним језиком. Стога, алгоритам моделовања тема није развијан од почетка већ се користило готово решење у оквиру софтверског пакета *Mallet*. Основни разлог развоја софтверског окружења налази се у потреби тестирања различитих претпоставки везаних за примену алгоритама моделовања тема у задатом проблему. Стога је развијено, у програмској језику Java, неколико класа које су имале са циљ обезбеђивање лаког и једноставног тестирања претпоставки.

Анализа резултата тестирања хипотеза вршила се помоћу Matlab-а и Microsoft Excel програма.

### 6.1 Општи преглед пакета Mallet

Софтверски пакет *Mallet* је софтвер **отвореног кода** - **енг. open source** и представља скуп алгоритама машинског учења оријентисаних на текстуалне податке. Обухвата различите врсте алгоритама за моделовања тема (**енг. topic modeling**), кластеровање (**енг. clustering**), класификацију (**енг. classification**), статистичке обраде природног језика (**енг. statistical natural language processing**) итд. Сви алгоритми развијени су у Java програмском језику

Алгоритам моделовања тема у овом пакету има две реализације. Основна верзија, која је коришћена у овом раду, представља имплементацију LDA-а (**енг. Latent Dirichlet Allocation**) Гибсовим узорковањем. Друга верзија представља хијерархијски LDA-а. Примена хијерархијске врста алгорита моделовања тема није била предмет овог рада али би било интересно испитати те могућности у будућем раду.

Софтверски пакет *Mallet* може се користити на два начина : конзолно - коришћењем предефинисаних команди, или се, обзиром на то да је код јавно доступан, сам код може убацити у постојећи пројекат. Пошто је циљ истраживања био специфичан и захтевао извесне модификације основне верзије решења у пакету *Mallet*, у раду је коришћена друга опција.

#### 6.1.1 Подаци у Mallet-у

*Mallet* користи објекте класе **Instance** за представљање података при чему сваки засебан објекат представља посебан документ. У конкретном случају под **документом** се подразумева текст питања или одговора. Дакле, један објект класе **Instance** представља или једно питање или један одговор.

Превођење „сирових“ података у објекте класе **Instance** представља предпроцесирање података за све алгоритме пакета *Mallet* и неопходан је корак при коришћењу било ког од тих алгоритама.

Превођење података у објекте класе **Instance**, обзиром на конструкторе, могло би да се посматра као тривијалан посао. Међутим, обзиром на специфичне захтеве исраживања, потребно је укључити и све кораке предпроцесирања који су наведени у претходном поглављу. За тако нешто је коришћена класа *Pipe*, такође из пакета *Mallet*.

*Pipe* је апстрактна класа и представља надкласу за све класе које се користе за модификовање објекта класе **Instance** у предпроцесирању. У пакету *Mallet* постоји неколико предефинисаних класа које врше неке од трансформација поменутих у претходном поглављу - нпр. уклањање HTML ознака (класа *CharSequenceRemoveHTML*), превођење у мала слова ( класа *CharSequenceLowercase()* ), формирање токена од текста (класа *CharSequence2TokenSequence* ) или уклањање често коришћених речи ( класа *TokenSequenceRemoveStopwords* ). Поред поменутих, већ уграђених класа, за потребе рада дописане су још и :

- класа за уклањање наставака речи - *PipeStem*
- класа за свођење на коренску реч - *StanfordLemmatizer*
- класа за убацивање синонима - *InsertSynonyms*

У додатку су дате неке од тих класа.

Пошто предпроцесирање најчешће захтева више од једне трансформације, пракса је формирање низа објеката типа *Pipe* који секвенцијално врше трансформацију објекта класе **Instance**. Тај низ најчешће се формира објектом класе *SerialPipe*.

Обзиром на то да улазни подаци у било који алгоритам машинског учења најчешће нису појединачни документи већ скупови докумената, од интереса је и класа *InstanceList* којом се једноставно апстрахује скуп докумената. Објекти ове класе садрже групу **Instance** објеката над којима су извршене исте трансформације (исти *SerialPipe*).

### 6.1.2 Алгоритам моделовања тема у Mallet-у

Алгоритам моделовања тема у Mallet-у представља имплементацију LDA-a (Latent Dirichlet Allocation) преко Гибсовог узорковања. Постоји више имплементација од којих је *ParallelTopicModel* најјефикаснија. У тој класи је реализован основни LDA-a алгоритам коришћењем Гибсовог узорковања. Перформансе су повећане паралелизацијом преко нити. Овај класа је коришћена у истраживању.

При моделовању тема следећи параметри су од интереса :

- број тема
- број итерација
- хиперпараметри

Вредности ових параметара се задају пре моделовања тема и од њих директно зависи решење. У истраживању највећи акценат је био на проналажењу оптималних вредности ових параметара и то пре свега броја тема и итерација. Вредности хиперпараметара су фиксирани тако да, што релистичније опишу знања о полазном скупу података. То подразумева подједнаку вероватноћу свих тема у оквиру докумената као и подједнаку вероватноћу свих речи у оквиру свих тема.

Приликом проналажење оптималне вредности броја тема и итерација прибегло се похлепном решењу тј. испитивању свих могућих комбинација. Разлог за такав приступ, поред проналажења оптималних вредности параметара, био је и испитивање **тренда** перформанси модела. Обзиром да је то временски изузетно захтеван посао, овај део истраживања одрађен је коришћењем **кластера**. У додатку је пример скрипте којом се покрећу послови на кластеру.

Процес моделовања тема извршава се у оквиру методе *estimate* класе *ParallelTopicModel* где се, у одређеном броју итерација, врши Гибсово узорковање по познатим обрасцима.



Овај процес још се назива и **тренирање модела**. Крајњи резултат представља **истрениран модел** који у себи садржи информације о :

- расподели тема унутар сваког документа
- расподели речи унутар сваке теме

Једна од интересантних ствари које се на основу ових информација могу закључити је и расподела тема унутар **новог** документа тј. документа који није коришћен при тренирању. То се постиже класом *TopicInferencer* која симулира у одређеном броју итерација претходни процес тренирања, стим што се промене односе само на нови документ. Дакле, овим није могуће накнадно тренирати већ истрениран модел, али је из података истренираног модела могуће наслутити расподелу тема на невидљивом документу.

Поред наведене класе, у Mallet-у постоји и *SimpleLDA* која такође имплементира LDA-а алгоритам коришћењем Гибсовог узорковања али у верзији која није оптимизирана. Због бољег разумевања суштине рада алгоритма, у почетној фази истраживања коришћена је ова класа. Касније се због перформанси прешло на решење у оквиру класе *ParallelTopicModel*.

## 6.2 Опис решења

Основни циљ развоја софтвера био је селектовање одговора који најбоље одговара на постављено питање. Идеја решења је изградња **модела тема** над свим одговорима чиме би се добио истрениран модел који поседује знање о расподели тема над документима као и расподели речи унутар тема. Затим се за постављено питање процењује расподела тема над њим и мери се **сличност** тог питања са свим одговорима у бази. Најсличнији одговор се проглашава одговором на постављено питање, при чему се кориснику предлаже 30 највероватнијих одговора.

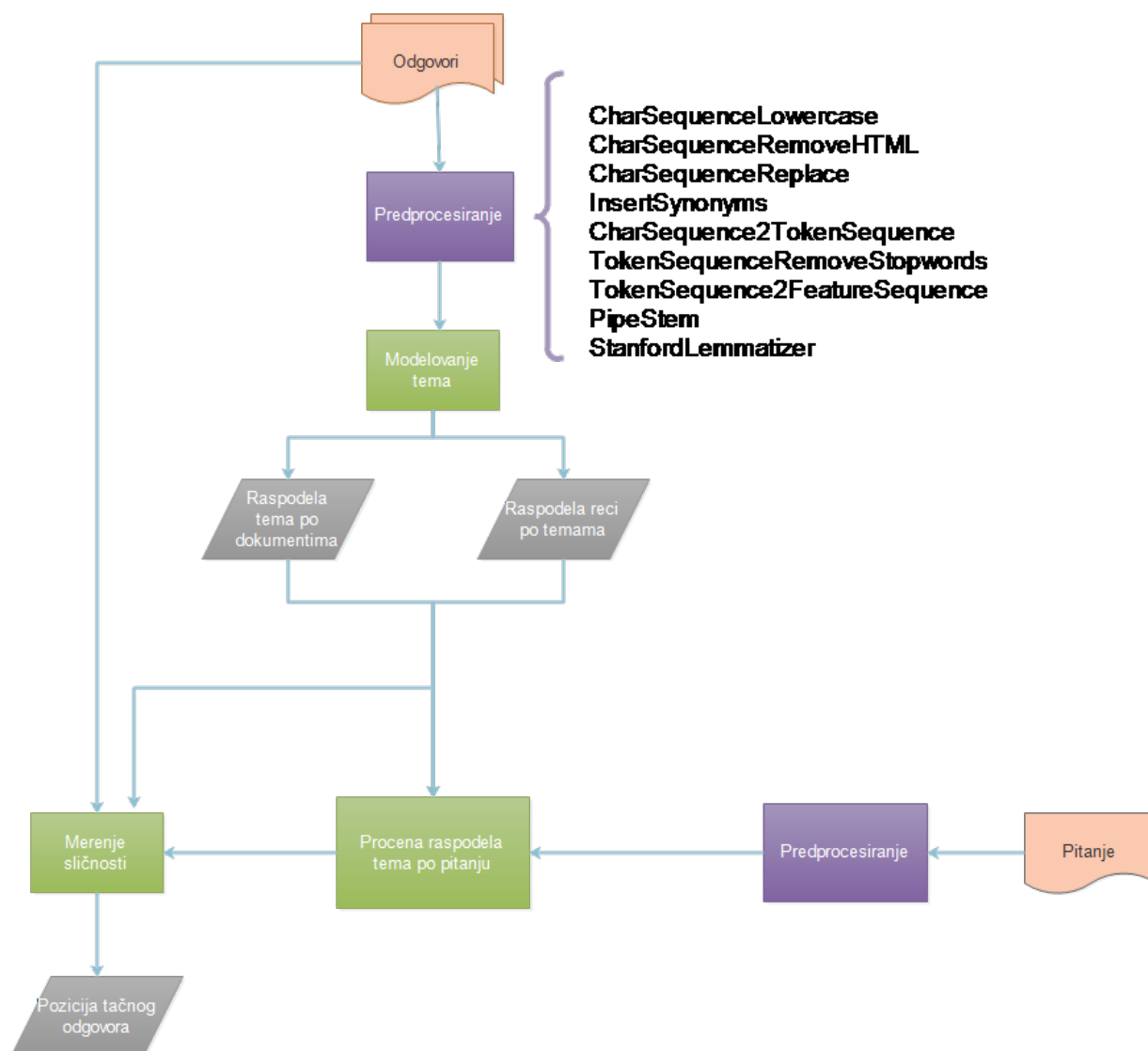
У улазне податке спадају листа питања и листа одговора на та питања. Дакле, зна се који одговор припада ком питању, односно може се установити да ли је програм адекватно проценио који је прави одговор. Стога се за оцену квалитета решења може узети **број тачних одговора** који су се јавили у првих 10, 20 или 30 предложених.

За мерење сличности питања и одговора коришћено је неколико метода :

- Косинусна сличност расподеле по темама - ово је најједноставнија мера сличности. Пошто је познат број тема, сваки документ се може представити вектором дужине броја тема. На  $i$  — месту у сваком вектору налази се вероватноћа, односно присуство  $i$  — теме у том документу. Косинусна сличност овакава два вектора представља меру сличности одговарајућих докумената. Што су расподеле по темама сличније, то ће мера сличности бити ближа 1. Основни разлог за овакво решење била је претпоставка да су питање и прави одговор **тематски** јако слични
- Сума косинусних сличности расподела речи - обзиром да је позната расподела речи по темама могуће је за сваку тему формирати вектор који за сваку реч из документа садржи вероватноћу те речи у одговарајућој теми. На овај начин се формирају два вектора, један за питање а други за одговор. Косинусна сличност ових вектора рачуна се за сваку тему а њихова сума представља меру сличности докумената
- Вероватноћа генерисања питања на основу одговора и њене варијације - обзиром да је позната расподела тема по документима као и расподела речи унутар тема, једноставно се може прерачунавати вероватноћа генерисања текста питања на основу текста одговора. Та вероватноћа представља меру сличности ова два документа. Варијације ове теме односе се на прерачунавање вероватноће. У раду су испитана још два додатна начина али је основна верзија показала најбоље резултате.

Испитане мере сличности разликују се по прецизности, брзини рада и меморијским захтевима. Испоставило се да су за примену на већи скуп података погодна прва и трећа (са варијантама) метода.

Идејни ток решења може се представити дијаграмом као на слици 3.1:



Слика 6.1: Ток решења

Важно је приметити да се у предпроцесирању не морају извршити све наведене трансформације нити у наведеном редоследу. Одабир подкупа трансформација директно утиче на резултат док редослед извршавања мора имати смислени ток. У конкретном истраживању, наведене трансформације су извршаване у датом редоследу. Главни циљ је био испитивање утицаја синонима, стеминга и лемитизације на резултат, тако да су ове три трансформације укључиване и искључиване како би се испитале све могуће комбинације.

Исто тако, ради конзистентности решења, структура и редослед предпроцесирања питања и одговора морају бити **исти**. Из тог разлога су на дијаграму и обојени истом бојом.

## Глава 7

# Решење проблема применом методе бројања речи

Метода бројања речи једна је од једноставнијих метода којом је могуће поредити два текстулана документа. Основна претпоставка ове методе је да су документи ближи један другом уколико имају више заједничких речи. Оваква претпоставка иако делује сасвим основано, не мора увек да буде тачна. Истим скупом речи могу се описати потпуно различите ствари и тиме генерисати два текстулана документа која, по смислу, уопште нису слична. Иако су овакви примери бројни, ова метода је, пре свега, због једноставности имплементације широко прихваћена у системима за проналажења одговора на постављено питање.

У конкретном раду, метода бројања речи је коришћена као **компаративно решење** у односу на решење применом алгоритма моделовања тема.

### 7.1 Опис решења методом бројања речи

Улаз у алгоритам су група питања и група одговора, при чему се за свако питање унапред зна који одговор из дата групе одговора представља тачан одговор. Решење методом бројања речи заснива се на мерењу сличности датог питања са **сваким одговором** у бази одговора. Након тога, одговори се рангирају према израчунатој сличности. Позиција тачног одговора у тој хијерархији свих одговора представља излаз који алгоритам даје за свако постављено питање.

Да би резултати ове методе могли да се пореде са резултатима претходно развијеног решења, неопходно је обезбедити **исте улазне податке**. Обизором на начин предпроцесирања у решењу базираном на моделовању тема, да би се обезбедили идентични улази коришћен је исти приступ предпроцесирању. То подразумева развој класе која, након предпроцесирања података за коришћење у алгоритму моделовања тема, те податке уписује на екстерни диск. Овим је обезбеђен апсолутно исти улаз и за компаративно решење.

Ради лакшег рачунања сличности докумената, сваки документ је представљен као вектор, тј. као низ неких нумеричких вредности. Трансформације текстуалног документа у вектор може се обавити на више начина. У раду су кориштена два приступа :

- једноставно бројање речи - Сваки документ представљен је као један низ. Свакој речи документа се додељује један природан број који представља индекс у том низу, при чему исте речи имају додељене исте бројеве. На тој позицији у низу налази се број појављивања те речи у документу. За мерење сличности два документа неопходно је обезбедити исто мапирање речи у природне бројеве. Ово значи да исти индекс у оба документа одговара истим речима.
- коришћење популарне TF-IDF методе - вектори се формирају на исти начин као код класичног приступа бројањем речи стим што се нумеричка вредност у вектору рачуна

по формулама TF-IDF методе.

За меру сличности два документа узета је **косинусна сличност** прерачунатих вектора.

## Глава 8

# Преглед резултата

Приликом израде рада развијено је софтверско окружење којим су се тестирали различити приступи решавању задатог проблема. Пре свега овде се мисли на различите мере сличности које су тестиране. Основна мера квалитета решења је **просечна позиција** тачног одговора у листи свих одговора. Што је просечна позиција нижа тј. ближа 1 то се решење сматра квалитетнијим, при чему се тежи да стандардна девијација буде што мања. Поред просечне позиције, битне карактеристике решења су брзина рада и меморијски захтеви.

Свака од тестираних мера сличности има своје параметре који су у мањој или већој мери осетљиви на одабир корака у предпроцесирању. Обзиром да се ради о алгоритму моделовања тема, у зависности од мере, биће потребан различит број тема и итерација како би се постигло оптимално решење. Због тога је било неопходно испитати за сваки меру посебно како који кораци предпроцесирања утичу и које вредности параметара алгоритма моделовања тема су оптималне за ту меру.

Прво тестирање решења проблема вршено је на скупу од 360 питања и 360 одговора преузетих са сајта *stackexchange.com*.

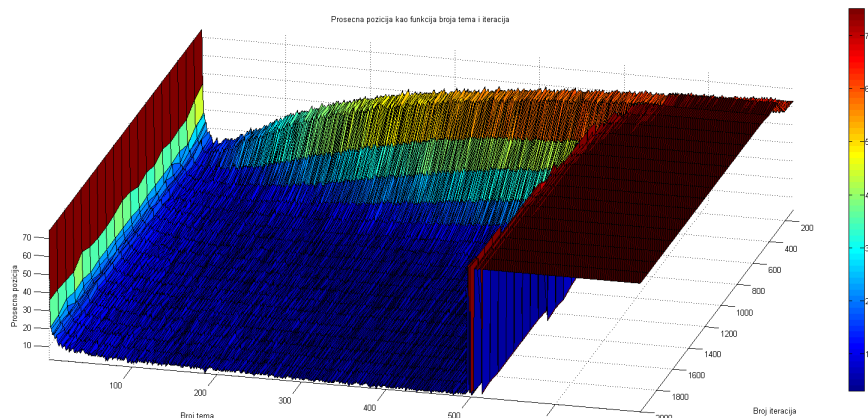
### 8.1 Утицај броја тема и броја итерација на просечну позицију

Обзором на то да расподела тема по документима, у мањој или већој мери, утиче на свако од тестираних мера, број тема представља битан параметар. Са малим бројем тема а на основу предложених мера, јако је тешко рангирати одговоре. Разлог томе је што ће се теме дефинисати према областима којима се баве документи. Према томе документи из исте области имаће сличне расподеле по темама а самим тим и приближно исту удаљеност од тестног документа односно питања. Због тога је неопходно да број тема буде довољно велики како би се овај проблем заобиша.

Међутим, није добро узети ни превелики број тема. У том случају сви документи ће имати сличну расподелу по темама па ће и одстојање од тестног документа бити приближно исто. Ово директно утиче на просечну позицију сводећи је на број који је једнак  $\lceil \frac{\text{број } Dokumenata}{2} \rceil$ . Овако нешто може се видети на следећем графику <sup>1</sup> :

---

<sup>1</sup>Ово тестирање је рађено на самом почетку и то на полазном скупу од 150 докумената. Тај скуп је подскуп скупа на коме су вршена сва остала тестирања



Слика 8.1: Зависност просечне позиције од броја тема и броја итерација

Прелаз на просечну позицију  $\lfloor \frac{\text{brojDokumenata}}{2} \rfloor$  (у конкретном случају 74 јер је број докумената 150) је прилично груб. Разлог томе је што нису тестиране све вредности броја итерација него свака 100-та. Како је за потребе рада било довољно уочити да се број тема не може бесконачно повећавати, ова појава даље није испитивана.

## 8.2 Утицај корака предпроцесирања на просечну позицију

Предпроцесирањем се **трансформише** текст улазних података како би био погоднији за обраду. Сама природа проблема намеће неке кораке предпроцесирања као неопходне. У њих спадају :

- превођење свих слова текста у мала слова - ова трансформација је неопходна како се не би реч написана почетним великим словом и иста реч написана почетним малим словом третирали као две различите речи
- уклањање HTML ознака - обзиром на порекло података, поред корисног текста, у документима се налазе и HTML ознаке које, са аспетка овог проблема, немају никаквог значаја.
- уклањање свих неалфанумеричких карактера - овом трансформацијом се уклањају знакови интерпункције и специјални знакови који нису битни за решење овог проблема
- уклањање често коришћених речи - оне не носе тематско значење, честе су у свим документима и представљају оптерећење при обради

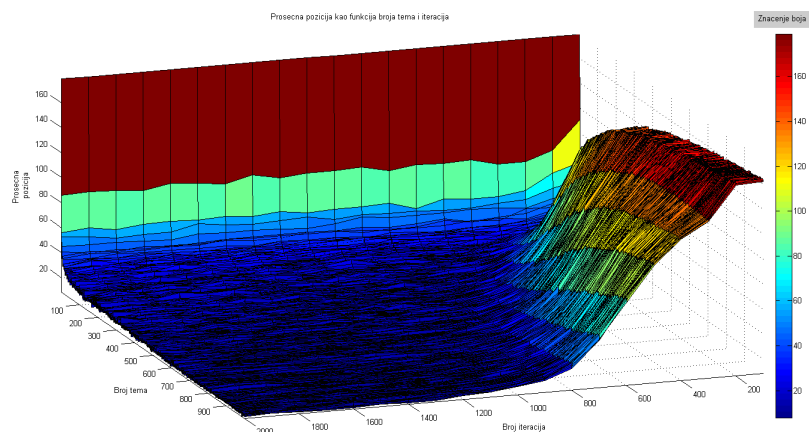
Ове трансформације су се увек укључивале, без обзира на меру сличности. Поред њих, додате су и стеминг, лемитизација и синоним трансформације у различитим комбинацијама.

### 8.2.1 Резултати без додатних трансформација

Улазни подаци - и питања и одговори - трансформисани преко три описана корака предпроцесирања и то у редоследу : превођење свих слова текста у мала слова, уклањање HTML ознака и уклањање често коришћених речи.

### 8.2.1.1 Косинусна сличност

Уколико се за меру сличности одабере **косинусна** мера, просечна позиција тачног одговора директно зависи од параметара модела тј. од броја тема и броја итерација. Та зависност се може приказати следећим графиком. ( слика 5.1)



Слика 8.2: Зависност просечне позиције од броја тема и броја итерација

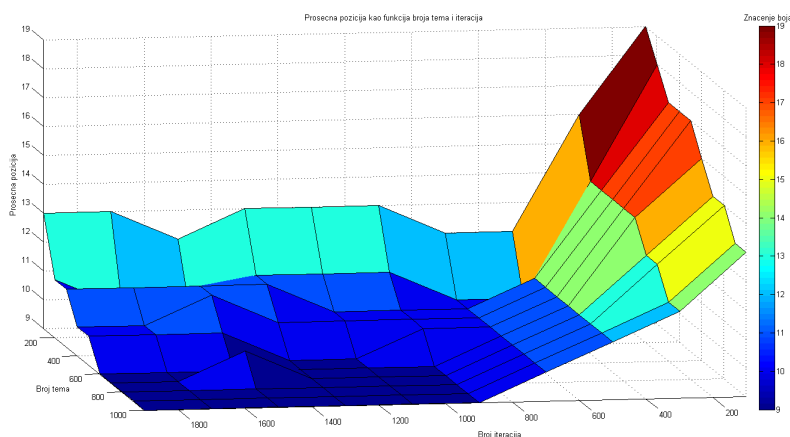
Минимална просечна позиција је **8** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.1: Утицај броја тема и итерација на просечну позицију тачног одговора

Број итерација	Број тема
1200	693
1400	756
1400	820

### 8.2.1.2 Мерење сличности према лексичкој и тематској сличности

Уколико се за меру сличности одабере мера **лексичке и теметске** сличности, просечна позиција такође директно зависи од параметра модела. Обзиром на временску и просторну сложеност ове мере, нису испитиване све могућности за параметре модела. График зависности просечне позиције од броја тема и итерација дат јена следећој слици (слика 5.2)



Слика 8.3: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **9** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.2: Утицај броја тема и итерација на просечну позицију тачног одговора

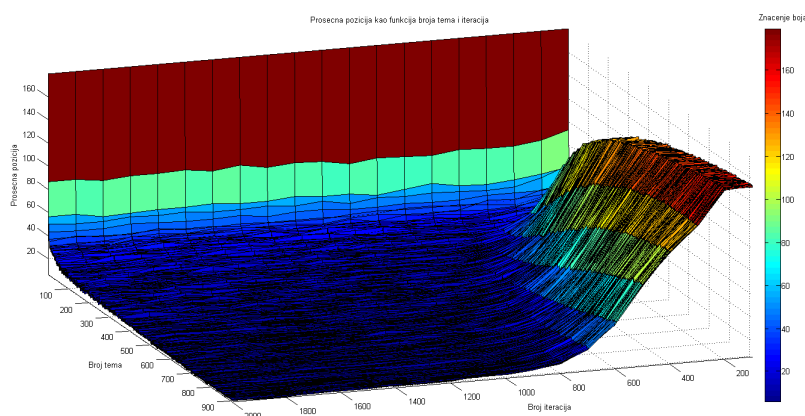
Број итерација	Број тема
900	600
900	700
1100	500

### 8.2.2 Утицај стеминга на резултат

Поред основне три трансформације, на улазне податке је примењене и стеминг модификација - уклањање наставака речи. Овим је полазни скуп података упрошћен јер је скуп различитих речи мањи. Без ове трансформације, једна иста реч у различитим родовима или временима се посматрала различито што је довело до великог диверзитета у скупу речи ( на пример речи енг. think и енг. thinking су посматране као различите речи без обзира на исто основно значење речи ).

#### 8.2.2.1 Косинусна сличност

Утицај стеминга на просечну позицију када се сличност мери косинусном сличношћу дат је на следећем графику( слика 5.2)



Слика 8.4: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **6** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.3: Утицај броја тема и итерација на просечну позицију тачног одговора

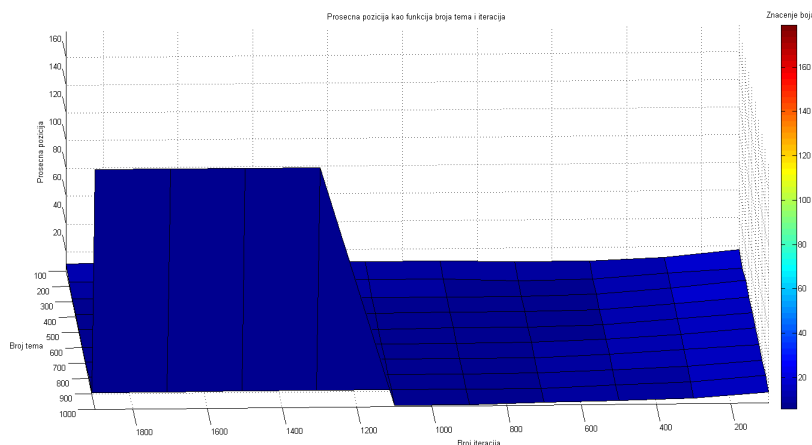
Број итерација	Број тема
1100	621
1200	643
1300	724

Додавање стеминга утиче на нижу просечну позицију и на упрошћавање модела. Без примене стеминга, најједноставнији модел којим се постиже оптимална просечна позиција је био комбинација 1200 - 693 док је са стемингом то 1100 - 621. Дакле, додавањем и стеминга коришћењем мањег броја тема и у мање итерација постижу се бољи резултати.



### 8.2.2.2 Мерење сличности према лексичкој и тематској сличности

Уколико се за меру сличности одабере мера **лексичке и теметске** сличности, просечна позиција такође директно зависи од параметра модела. Обзиром на временску и просторну сложеност ове мере, нису испитиване све могућности за параметре модела. График зависности просечне позиције од броја тема и итерација дат јена следећој слици (слика 5.2)



Слика 8.5: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **6** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.4: Утицај броја тема и итерација на просечну позицију тачног одговора

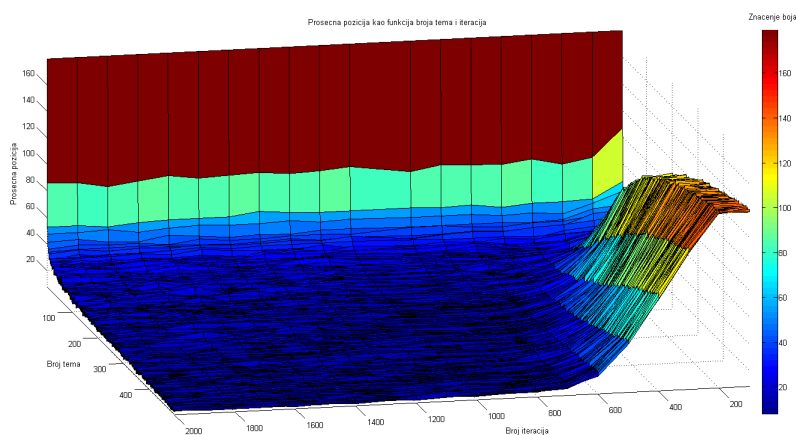
Број итерација	Број тема
1100	500
1300	600
1300	800

### 8.2.3 Утицај лемитизације на резултат

Лемитизација је процес свођења речи на **коренску** реч. Склањањем наставака речи могуће је пронаћи заједничку основу речи. Али, уколико реч мења облик у различитим лицима или временима, склањањем наставка речи, уколико уопште постоје, неће се добити иста реч. Лемитизацијом се овај проблем решава. На основу граматичких правила, препознаје се који основни облик реч и тим обликом замњује сепојављивање те речи у било којој форми.

#### 8.2.3.1 Косинусна сличност

Утицај лемитизације на просечну позицију када се сличност мери косинусном сличношћу дат је на следећем графику( слика 5.2)



Слика 8.6: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је 8 и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.5: Утицај броја тема и итерација на просечну позицију тачног одговора

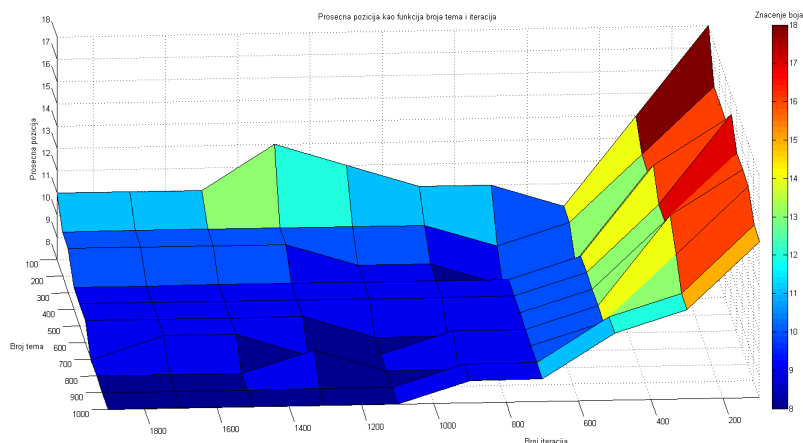
Број итерација	Број тема
1300	481
1300	486
1500	481
1600	477

Може се закључити да лемитизација значајно утиче на поједностављивање модела. Најмања вредност просечне позиције добије се коришћењем броја тема око 470 за разлику од основног модела где тај број износи преко 600.

Обзиром на то да лемитизација своди на коренску реч, мањи је диверзит скупа речи. Самим тим, моделу је теже да препозна прави одговор па је због тога просечна позиција више него просечна позиција која се добија употребом стеминга.

### 8.2.3.2 Мерење сличности према лексичкој и тематској сличности

Уколико се за меру сличности одабере мера **лексичке и теметске** сличности, просечна позиција такође директно зависи од параметра модела. Обзиром на временску и просторну сложеност ове мере, нису испитиване све могућности за параметре модела. График зависности просечне позиције од броја тема и итерација дат јена следећој слици (слика 5.2)



Слика 8.7: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је 8 и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.6: Утицај броја тема и итерација на просечну позицију тачног одговора

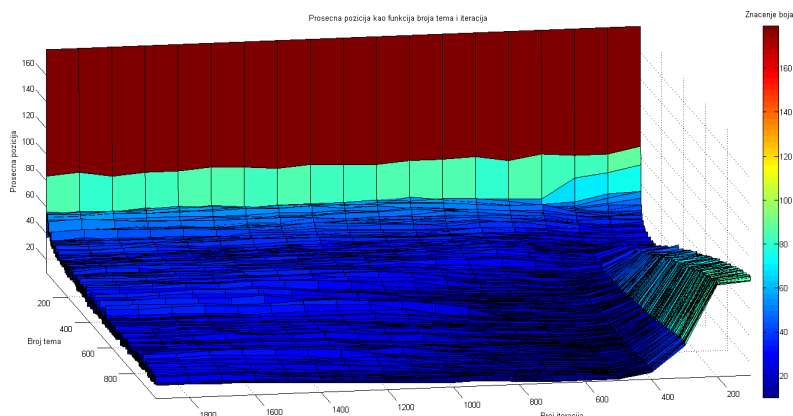
Број итерација	Број тема
700	300
900	600
900	800

#### 8.2.4 Утицај додавања синонима на резултат

Додавање синонима речи има за циљ боље тематско одвајање. Наиме, уколико се свакој речи придода неколико синонима, претпоставља се да ће тематско одвајање бити једноставније. Основни разлог за увођење ове врсте трансформација била чињеница да човек поузданије закључује „о чему се ради“ у неком тексту уколико му се наведе неколико синонима за кључне речи. Обзиром да се ниједна реч не потенцира свакој речи је додат неки број синонима. Будући да додавање синонима директно утиче на перформансе система, у конкретном раду је одлучено да тај број буде 5.

##### 8.2.4.1 Косинусна сличност

Утицај на додавања синонима на просечну позицију када се сличност мери косинусном сличношћу дат је на следећем графику( слика 5.2)



Слика 8.8: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **10** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

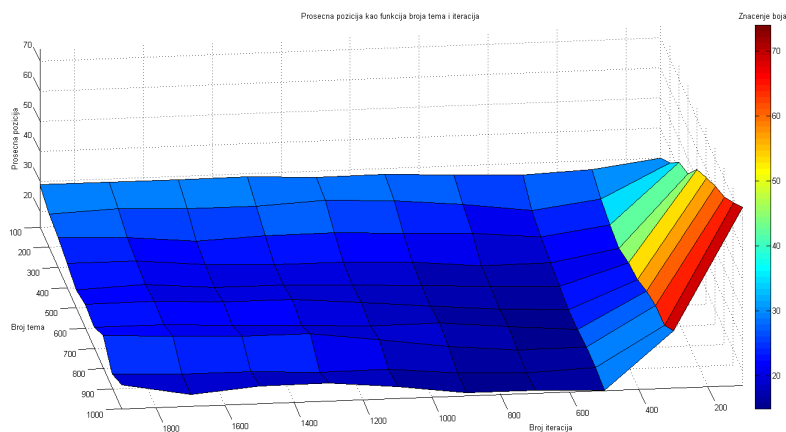
Табела 8.7: Утицај броја тема и итерација на просечну позицију тачног одговора

Број итерација	Број тема
500	674
500	675
600	773
500	991

На основу облика графика може се закључити да додавање синонима утиче на екстремне вредности просечне позиције. Такође, упрошћава се модел и у мањем броју итереације достиже се минимална вредност просечне позиције. Са друге стране, пуно речи имају другачије значење у различитим контекстима а обзиром да у систем није убачено никакво додатно знање, постоји опасност од додавања неадекватних речи. То може да се одрази да немогућност прецизног одређивања тематске слике документа па и на прецизност решења.

#### 8.2.4.2 Мерење сличности према лексичкој и тематској сличности

Уколико се за меру сличности одабере мера **лексичке и теметске** сличности, просечна позиција такође директно зависи од параметра модела. Обзиром на временску и просторну сложеност ове мере, нису испитиване све могућности за параметре модела. График зависности просечне позиције од броја тема и итерација дат јена следећој слици (слика 5.2)



Слика 8.9: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **15** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.8: Утицај броја тема и итерација на просечну позицију тачног одговора

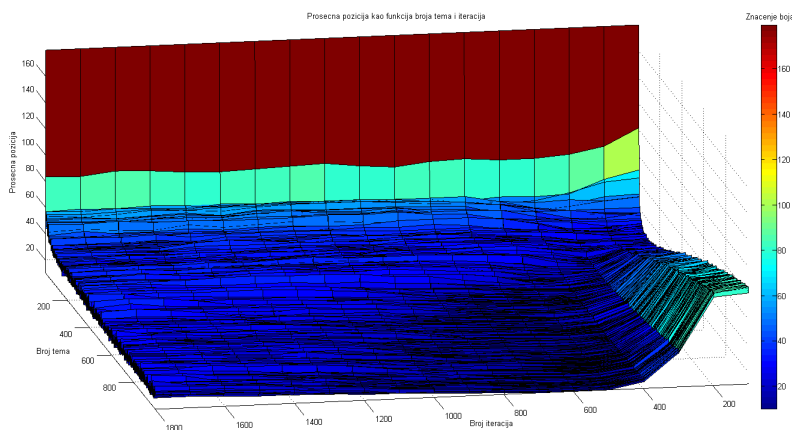
Број итерација	Број тема
500	600
500	700
500	1000

## 8.2.5 Укупни резултати са синонимима и стемингом

Поред појединачане примене сваке трансформације на улазне податке, у раду су тестиране и њихове комбинације. Обзором на то да стеминг уклања наставке, велики број тако модификованих добија нерегуларни облик. Због тога није могуће ни пронаћи њихове синониме. Да би се то избегло, најпре су додавани синоними за сваку реч а затим је извршен стеминг.

### 8.2.5.1 Косинусна сличност

Утицај комбинације додавања синонима и стеминга на просечну позицију када се сличност мери косинусном сличношћу дат је на следећем графику( слика 5.2)



Слика 8.10: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **10** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.9: Утицај броја тема и итерација на просечну позицију тачног одговора

Број итерација	Број тема
500	555
500	690
500	712
700	847

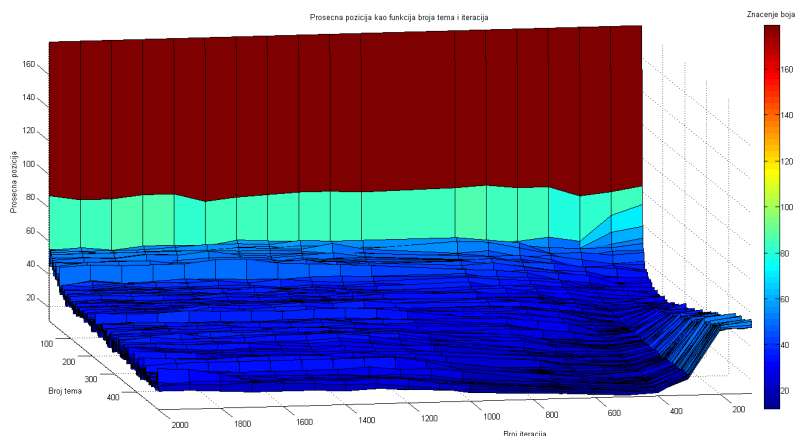
Са графика је јасно да је утицај синонима значајан. Облик површи је јако сличан облику који се добија додавањем само синонима, без примене стеминга. Дакле, додавање синонима има јачи утицај од стеминга. Односно, додавање стеминг трансформације након убацивања синонима не утиче значајно на просечну позицију.

## 8.2.6 Укупни резултати са синонимима и лемитизацијом

Још једна од комбинација улазних трансформација која је тестирана у овом раду је и комбинација лемитизација и синоними. Лемитизација ће свести речи на коренску, без обзира на облик те речи. Међутим, поред означавања времена и лица, различит облик речи може означавати и различито значење речи. Из тог разлога, најпре су додати синоними па је затим примењена лемитизација.

### 8.2.6.1 Косинусна сличност

Утицај комбинације додавања синонима и лемитизације на просечну позицију када се сличност мери косинусном сличношћу дат је на следећем графику( слика 5.2)



Слика 8.11: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **12** и добија се при тачно једној комбинацији параметара и то :

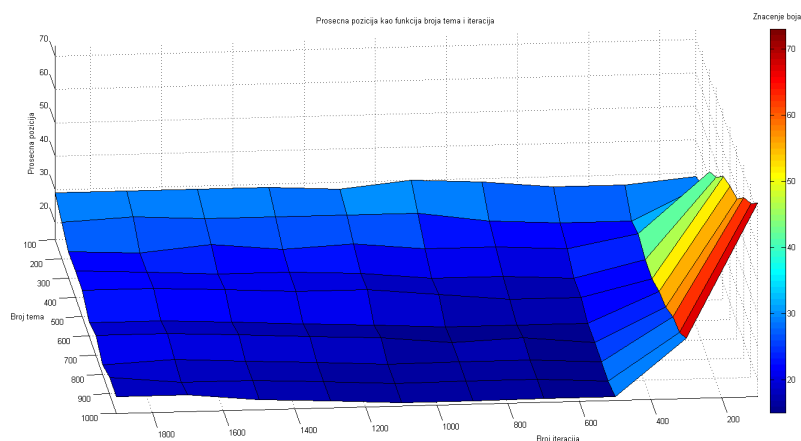
Табела 8.10: Утицај броја тема и итерација на просечну позицију тачног одговора

Број итерација	Број тема
600	483

Поново се примећује знатан утицај синонима на облик површи као и на просечну позицију. Лемитизација није значајно утицала на просечну позицију.

#### 8.2.6.2 Мерење сличности према лексичкој и тематској сличности

Уколико се за меру сличности одабере мера **лексичке и теметске** сличности, просечна позиција такође директно зависи од параметра модела. Обзиром на временску и просторну сложеност ове мере, нису испитиване све могућности за параметре модела. График зависности просечне позиције од броја тема и итерација дат јена следећој слици (слика 5.2)



Слика 8.12: Зависност просечне позиције од броја тема и броја итерација

Минимална просечна позиција је **15** и добија се при више различитих комбинација параметара. Неке од комбинација су приказане у следећој табели

Табела 8.11: Утицај броја тема и итерација на просечну позицију тачног одговора

Број итерација	Број тема
500	600
500	700
500	800

### 8.3 Упоредни резултати решења алгоритмом моделовања тема и бројањем речи

#### 8.3.1 Косинусна сличност

Табела 8.12: Упоредни резултати алгоритме моделовања тема и решења бројањем речи

	WC				TM			
	Prva	Top 10	Top 20	Top 30	Prva	Top 10	Top 20	Top 30
noStemNoSyn	231	324	339	343	179	288	308	324
stemNoSyn	233	331	345	350	190	296	319	330
lemmNoSyn	231	333	345	350	187	297	329	336
stemSyn	177	319	337	342	147	282	304	324
lemmSyn	175	312	335	342	146	280	309	322
noStemSyn	175	309	335	339	147	286	305	325

Из резултата се може уочити да се косинусном мером расподела тема по документима не могу добити бољи резултати од класичног приступа бројањем речи. Разлог за то може да буде недовољна количина података али и природа мера сличности. Обзором да се узима само тематска сличност, занемарује се јако битна карактеристика сличности два документа - лексичка.

Добијени резултати, обзором да се ради о малом скупу података, могу да наведу на закључак да постоји извесна зависност између дужине питања ( одговора) и датих решења.

#### 8.3.2 Мерење сличности према лексичкој и тематској сличности

Обзиром на то да је ова сличност јако велике сложености, из практичних разлога, нису тестиране све трансформације. Резултати који су добијени дати су у следећој табели:

Табела 8.13: Упоредни резултати алгоритме моделовања тема и решења бројањем речи

	WC				TM			
	0	10	20	30	0	10	20	30
noStemNoSyn	231	324	339	343	217	308	332	340
stemNoSyn	233	331	345	350	215	325	339	343
lemmNoSynEnd	231	333	345	350	213-232	319	338	342
lemmSyn	175	312	335	342	150	270	288	304

Дакле, ова мера показује незнатно слабије резултате од класичног метода, стим што јој је комплексност знатно већа. Са друге стране, добијена прецизност решења је већа него код косинусне сличности. Међутим, како је потребно доста времена да би се добили резултати, ова мера није даље разматрана.



### 8.3.3 Сличност према предвиђеној вероватноћи

Полазни тестни скуп има 360 питања и 360 одговора. Он је довољно мали да омогућава тестирање сваке од поменутих мера, довољно велики да би се уочили трендови али недовољан да би се донели генерални закључци.

Обзиром на то да је ова мера показала најбоље резултате, поред стандардног тестног скупа, тестирана је и на већим скуповима података. Величине додатних скупова података су по 200, 800, 1400, 2000, 10000, 20000 и 40000 докумената, односно 100, 400, 700, 1000, 5000, 10000 и 20000 питања и исто толико одговора. Подаци су преузети са сајта *answers.yahoo.com*

Ова додатна тестирања су била неопходна како би се избегао енг. *overfitting*, тј. како би се донели закључци који што мање зависе од података.

Енг. *overfitting* је термин машинског учења који се односи на прилагођавање програма подацима. Тада се најчешће дешава да програм са изузетно великом прецизношћу ради са једним скупом података, док са другим скупом та прецизност драстично опада. Разлог томе је што је програм **научен** да ради само са одређеним скупом података. У конкретном случају, немогуће је потпуно побећи од енг. *overfitting*-а због природе проблема. Међутим, уколико се покаже да модели грађени над различитим скупом података показују исте особине, тада се могу извести закључци који су релативно независни од података.

Компаративни модел, као и досада било је решење бројањем речи. Резултати добијени мерењем дати су у следећој табели :

Табела 8.14: Упоредни резултати решења алгоритма моделовања тема употребом мере сличности на основу предвиђене вероватноће и компаративног решења

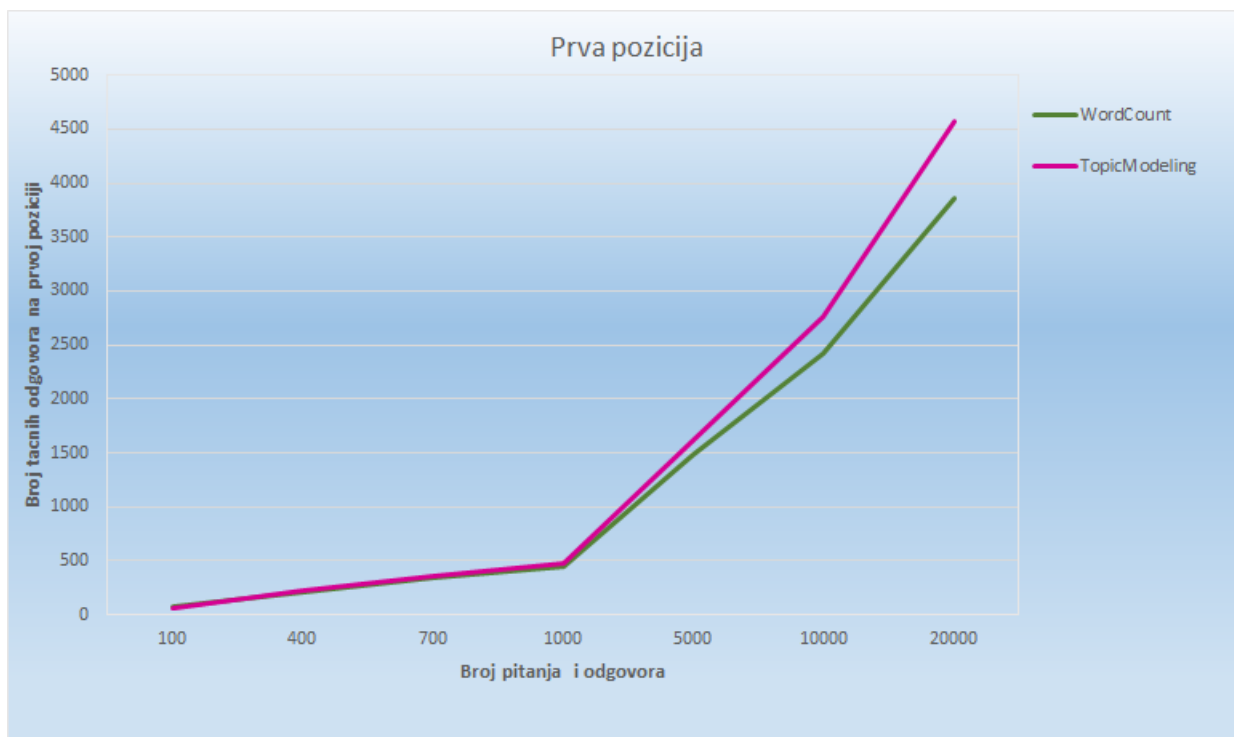
	WC				TM				
Broj dokumenata	Prva	Top 10	Top 20	Top 30	Prva	Top 10	Top 20	Top 30	Br. tema
100	72	90	91	91	66	87	91	94	50
400	203	338	353	361	217	326	336	348	400
700	338	569	596	610	357	546	583	594	750
1000	453	789	836	853	472	744	794	829	1100
5000	1484	3002	3367	3570	1614	2960	3279	3472	800
10000	2422	5334	6054	6521	2766	5297	5988	6421	1100
20000	3866	9044	10516	11460	4576	9182	10546	11358	900

Као што се може приметити, алгоритам моделовања тема готово увек успева да на прву позицију избаци више тачних одговора него компаративно решење. Једино за улазни скуп од 100 питања и 100 одговора ово није испуњено, иако не одступа много. Може бити више разлога за овакво понашање:

- Природа података које погодује компаративном решењу
- Недовољна количина података - модел нема довољно информација

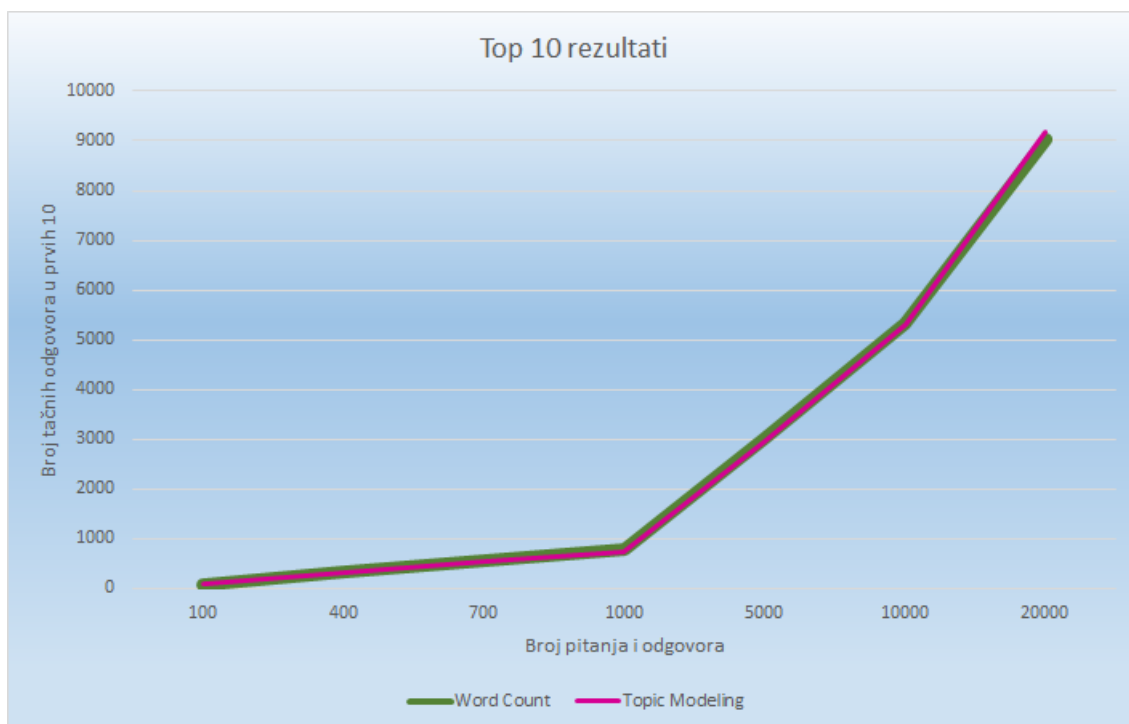
Даље испитивање узрока оваквог понашања није од интереса. Пре свега, улазни скуп је јако мали, тако да резултати доста зависе од података. Са друге стране, ова величина скупа је недовољна за било какву реалну примену.

Уколико се графички представи број докумената који се појављују на првој позицију у односу на величину улазног скупа, може се приметити да са повећањем улазних података, разлика у резултатима ова два решења постаје већа. Графички приказ дат је на следећој слици :



Слика 8.13: Прва позиција у односу на величину улазног скупа

Појављивање тачног одговора на првој позицији може бити престрог критеријум за упоређивање резултате ове две методе. Стога се поред прве позиције, за меру оцене може узети и појављивање тачног одговора у првих 10 предложених резултата. Графички приказ резултата поменуће две методе дат је на следећој слици.

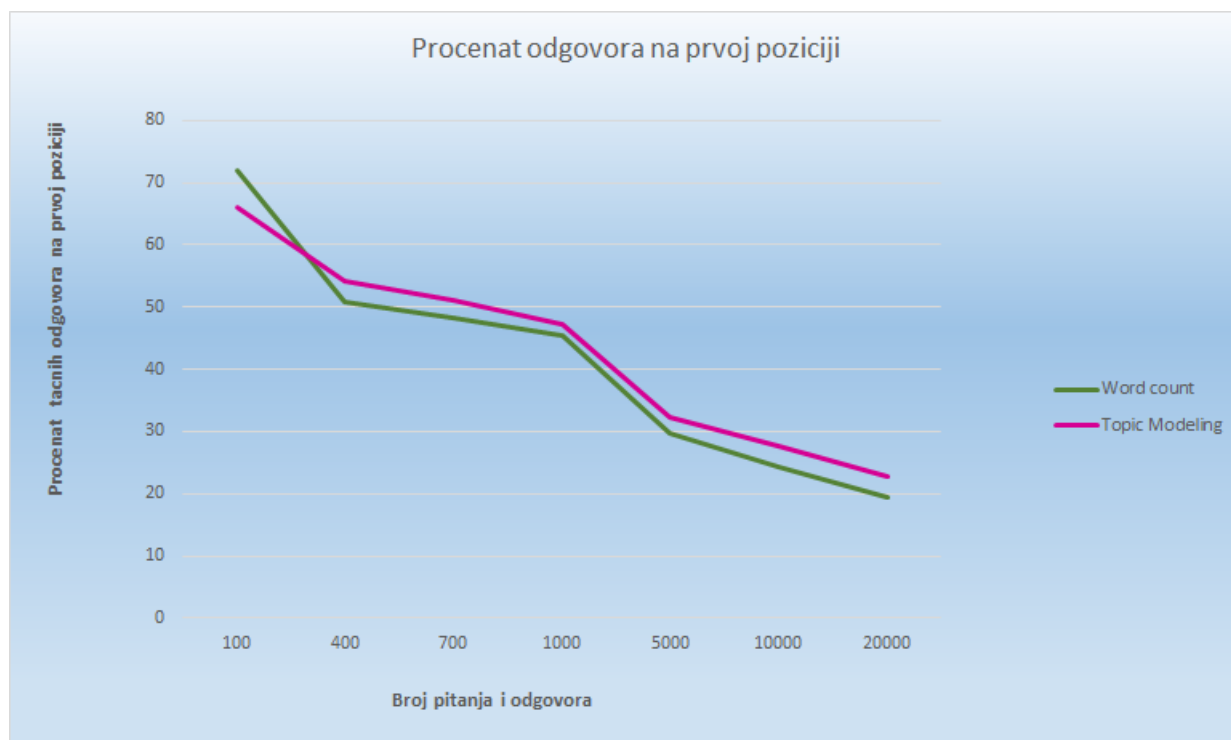


Слика 8.14: Број тачних одговора у првих 10 предложених резултата у односу на величину улазног скупа

Као што се може уочити са слике, резултати ове две методе мерење са становишта првих 10 предложених резултата, готово се идентично понашају.

Поред броја тачних одговора који се налазе на првој позицији, или у првих 10 предложених одговора, важна карактеристика оба решења је **прецизност**. Под прецизношћу решења подразумева се **проценат** тачних одговора који испуњавају неки критеријум. У конкретном случају, од интереса је да ли се налазе на првој позицији, односно да ли се налазе у првих 10 предложених решења.

Прецизност оба решења у односу на то да ли се тачан одговор појављује на првој позицији у листи предложених одговора дат је на следећој слици :

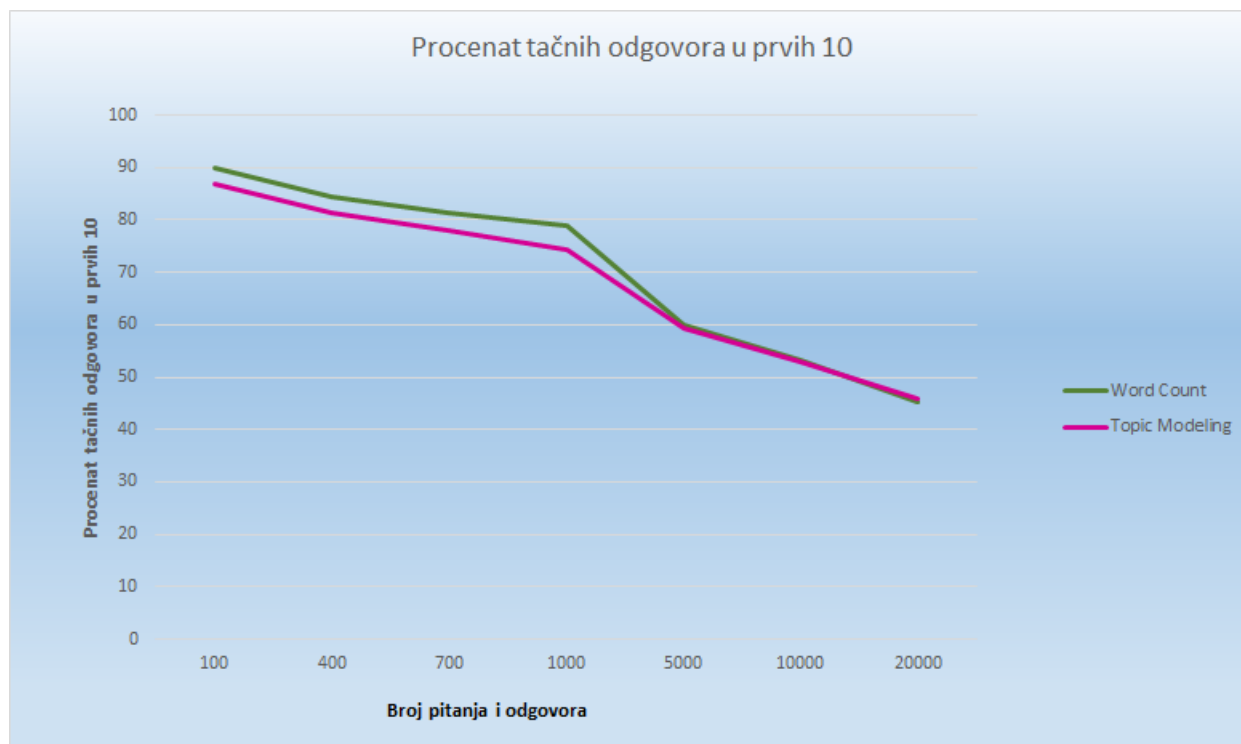


Слика 8.15: Прецизност предложених решења

Са слике се може закључити следеће :

- Оба решења губе прецизност са повећањем броја улазних података
- Прецизност решења алгоритмом моделовања тема спорије опада него прецизност решења бројањем речи
- Разлика у прецизности ова два решења повећава се са повећањем броја улазних података

Упоредна прецизност предложених решења у односу на то да ли се тачан одговор налази у првих 10 предложених одговора, дата је на следећем графику :



Слика 8.16: Прецизност предложених решења

Могу се извести слични закључци као и за прецизност везану за појављивање тачног одговора на првој позицији :

- Са порастом величине улазних података, прецизност оба решења опада
- Прецизност решења бројањем речи је већа него прецизност решења моделовањем тема за величину улазног скупа до 5000. Након тога, прецизности оба решења су готово исте.

### Статистичка значајност разлике у резултатима

Како би изведени закључци били поузданији, урађени су и статистички тестови значајности над добијеним резултатима. У конкретном случају рађен је Вилкоксонов тест за упоредне резултате добијене на првој позицији оба решења. Овим тестом се испитује да ли постоји статистички значајна разлика у резултатима ове две методе. Услов значајности је да сигнификантност добијена овим тестом буде  $< 0.05$ . Резултати тестова налазе се у следећој табели :

Табела 8.15: Резултати Вилкоксоновог теста

Broj dokumenata	Prva WC	Prva TM	Signifikantnost	Značajno
100	72	66	0.7109999999999997	NE
400	203	217	3.1E-2	DA
700	338	357	2.1000000000000001E-2	DA
1000	453	472	2E-3	DA
5000	1484	1614	7.499999999999997E-2	NE
10000	2422	2766	2.7E-2	DA
20000	3866	4576	0	DA

На основу резултата у претходној табели, може се закључити да разлике у мерењима нису случајне већ зависне од метода. При тестирању резултата на скупу од 5000 одговора, утврђено је да разлике између метода нису статистички значајне. Ово не мора да значи да разлике између метода не постоје. Праг значајности са којим су рађени тестови је 0.05, док је сигнификантност добијена у овом скупу података 0.07. Ова мала разлика може да буде последица од грешке у рачуну ( нагомилана грешка ) до специфичности података.

## Глава 9

# Додатак

### 9.1 Предпроцесирање

```
1
2 public class PipeStem extends Pipe{
3
4     private static final long serialVersionUID = 1L;
5
6     public Instance pipe(Instance carrier) {
7         SnowballStemmer stemmer = new englishStemmer();
8         TokenSequence in = (TokenSequence) carrier.getData();
9
10        for (Token token : in) {
11            stemmer.setCurrent(token.getText());
12            stemmer.stem();
13            token.setText(stemmer.getCurrent());
14        }
15
16        return carrier;
17    }
18
19 }
```

---

```
1
2
3 public class StanfordLemmatizer extends Pipe{
4
5     protected StanfordCoreNLP pipeline;
6
7     public StanfordLemmatizer() {
8         Properties props;
9         props = new Properties();
10        props.put("annotators", "tokenize, ssplit, pos, lemma");
11
12        this.pipeline = new StanfordCoreNLP(props);
13    }
14
15 public Instance pipe(Instance carrier) {
16
17     TokenSequence in = (TokenSequence) carrier.getData();
18
19     for (Token token : in) {
20         String text = lemmatize(token.getText());
21         token.setText(text);
22     }
23
24     return carrier;
```

```

25     }
26
27     public String lemmatize(String documentText)
28     {
29         List<String> lemmas = new LinkedList<String>();
30         Annotation document = new Annotation(documentText);
31         this.pipeline.annotate(document);
32         List<CoreMap> sentences = document.get(SentencesAnnotation.class);
33         String lemmasString = "";
34         for(CoreMap sentence: sentences) {
35             for (CoreLabel token: sentence.get(TokensAnnotation.class)) {
36                 lemmasString+=token.get(LemmaAnnotation.class);
37             }
38         }
39         return lemmasString;
40     }
41
42
43
44 }

```

---

## 9.2 Оптималан број тема и итерација

```

1
2 for i in `seq 100 200 2000`
3 do
4     qsub -v topic=500,iter=$i job.sub
5 done

```

---

job.sub :

```

1
2 java -Xms6000m -Xmx10000m -classpath "/lustre/home/jvasiljevic/
   TopicModeling/Mallet/class:/lustre/home/jvasiljevic/TopicModeling/
   Mallet/lib/mallet-deps.jar:/lustre/home/jvasiljevic/TopicModeling/
   Mallet/lib/jdom-1.0.jar:/lustre/home/jvasiljevic/TopicModeling/Mallet/
   lib/grmm-deps.jar:/lustre/home/jvasiljevic/TopicModeling/Mallet/lib/
   weka.jar:jaws-bin.jar:./stanford-lemmitization/stanford-corenlp-3.5.2.
   jar:./stanford-lemmitization/stanford-corenlp-3.5.2-models.jar:." -
   Dwordnet.database.dir=/usr/share/wordnet-3.0/dict/
   TestFixParamKlaster ../../odgovor1.csv ../../pitanje1.csv $lambda $mi
   ../../Mallet/stoplists/stopwords.txt

```

---

## 9.3 Мерење сличности

```

1
2 public static double dotProd(double[] a, double[] b){
3     if(a.length != b.length){
4         throw new IllegalArgumentException("The dimensions have to be equal!");
5     }
6     double sum = 0;
7     for(int i = 0; i < a.length; i++){
8         sum += a[i] * b[i];
9     }
10    return sum;
11 }
12 public static double intensity(double[] a)
13 {
14     double sum = 0;
15
16     for(int i = 0; i < a.length; i++){

```

```
17         sum+= a[i]*a[i];
18     }
19
20     return Math.sqrt(sum);
21 }
22
23
24
25
26 public static double cosineSim(double[] broj1,double[] broj2)
27 {
28     return dotProd(broj1, broj2)/(intensity(broj1)*intensity(broj2));// *
        intensity0dg(broj2,broj1));
29 }
```

---



## Глава 10

## Литература

# Литература

- [1] Lars Vogel, Android Service and Broadcast Receiver, [www.vogella.de](http://www.vogella.de), 2011.
- [2] David M. Blei, Introduction to Probabilistic Topic Models, Princeton University
- [3] David M. Blei, Topic Models, Princeton University, September 1, 2009
- [4] David Mimno, The details: training and validating big models on big data, Princeton University
- [5] Ivana Kovačević, Verovatnoća i statistika sa zbirkom zadataka, Beograd 2011.
- [6] Violeta Aleksić, Elementi teorije verovatnoće i matematičke statistike,
- [7] [www.ekfak.kg.ac.rs](http://www.ekfak.kg.ac.rs), kurs Osnovi statistike, avgust 2015.
- [8] <http://starisajt.elfak.ni.ac.rs/phptest/new/html/Studije/predavanja-literatura/matematika-odabrana-poglavlja/verovatnoca.pdf>, avgust 2015
- [9] Random Signals and Processes Primer with MATLAB, Gordana Jovanovic Dolocek, 2013
- [10] Lecture 32: Markov Chains Continued | Statistics 110 on youtube - <https://www.youtube.com/watch?v=aBGOyZv2pZE>, avgust 2015
- [11] Christopher M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006
- [12] <http://fourier.eng.hmc.edu/e161/lectures/gaussianprocess/node7.html>, avgust 2015
- [13] <https://theclevermachine.wordpress.com/2012/11/05/mcmc-the-gibbs-sampler/>, avgust 2015
- [14] William M. Darling, A Theoretical and Practical Implementation Tutorial on Topic Modeling and Gibbs Sampling, December 1, 2011
- [15] Yi Wang, Distributed Gibbs Sampling of Latent Topic Models: The Gritty Details, August , 2008
- [16] <http://mallet.cs.umass.edu> , Avgust 2015
- [17] LDA Based Similarity Modeling for Question Answering, Asli Celikyilmaz, Dilek Hakkani-Tur, Gokhan Tur

УНИВЕРЗИТЕТ У КРАГУЈЕВЦУ  
ПРИРОДНО-МАТЕМАТИЧКИ ФАКУЛТЕТ  
ИНСТИТУТ ЗА МАТЕМАТИКУ И ИНФОРМАТИКУ

Завршни рад под називом  
одбраћен је \_\_\_\_\_.

МЕНТОР:

\_\_\_\_\_  
др Име Презиме, звање, Институција

ЧЛАНОВИ КОМИСИЈЕ:

\_\_\_\_\_  
др Име Презиме, звање, Институција

\_\_\_\_\_  
др Име Презиме, звање, Институција

Завршни рад је оцењен оценом \_\_\_\_\_.